

|              |                                                                               |
|--------------|-------------------------------------------------------------------------------|
| Title        | 胚発生のパターン形成の仕組みの創発に関わる研究                                                       |
| Author(s)    | 萩原, 茂                                                                         |
| Citation     |                                                                               |
| Issue Date   | 2000-03                                                                       |
| Type         | Thesis or Dissertation                                                        |
| Text version | author                                                                        |
| URL          | <a href="http://hdl.handle.net/10119/658">http://hdl.handle.net/10119/658</a> |
| Rights       |                                                                               |
| Description  | Supervisor:小長谷 明彦, 知識科学研究科, 修士                                                |

修 士 論 文

**胚発生のパターン形成の仕組みの創発に関わる研究**

指導教官 小長谷 明彦 教授

北陸先端科学技術大学院大学  
知識科学研究科知識システム基礎学専攻

萩原 茂

平成12年2月15日

# 目次

|          |                                                           |           |
|----------|-----------------------------------------------------------|-----------|
| <b>1</b> | <b>はじめに</b>                                               | <b>8</b>  |
| 1.1      | 目的 . . . . .                                              | 8         |
| 1.2      | 背景 . . . . .                                              | 8         |
| 1.3      | 実験と結果 . . . . .                                           | 9         |
| 1.4      | 実験環境の構築 . . . . .                                         | 10        |
| 1.5      | 本論文の構成 . . . . .                                          | 10        |
| <b>2</b> | <b>生物学的背景</b>                                             | <b>11</b> |
| 2.1      | ショウジョウバエの胚発生の概略 . . . . .                                 | 11        |
| 2.2      | 胚の前後部軸のパターン形成モデル . . . . .                                | 12        |
| 2.2.1    | 母系効果遺伝子 . . . . .                                         | 13        |
| 2.2.2    | 分節遺伝子 . . . . .                                           | 14        |
| 2.3      | まとめ . . . . .                                             | 20        |
| <b>3</b> | <b>関連する研究</b>                                             | <b>22</b> |
| 3.1      | SIMFLY および SIMFLY2 . . . . .                              | 22        |
| 3.2      | “Simulation of <i>Drosophila</i> Embryogenesis” . . . . . | 24        |
| 3.3      | 関連する研究についての議論 . . . . .                                   | 25        |
| <b>4</b> | <b>縞形成の仕組みの進化シミュレーション</b>                                 | <b>26</b> |
| 4.1      | はじめに . . . . .                                            | 26        |
| 4.2      | 目的 . . . . .                                              | 26        |
| 4.3      | 方法 . . . . .                                              | 27        |
| 4.3.1    | プログラムで使用されているモデル . . . . .                                | 27        |

|          |                                       |           |
|----------|---------------------------------------|-----------|
| 4.3.2    | 個体の評価方法 . . . . .                     | 28        |
| 4.3.3    | プログラムの実行の大まかな流れ . . . . .             | 29        |
| 4.3.4    | 使用している記号 . . . . .                    | 31        |
| 4.3.5    | 使用している遺伝的オペレータ . . . . .              | 32        |
| 4.4      | 実験結果 . . . . .                        | 33        |
| 4.4.1    | 実験 1 . . . . .                        | 33        |
| 4.4.2    | 実験 2 . . . . .                        | 34        |
| 4.4.3    | 実験 3 . . . . .                        | 39        |
| 4.4.4    | 実験 1, 2, 3 についてのまとめ . . . . .         | 50        |
| 4.4.5    | 実験 4 . . . . .                        | 51        |
| 4.5      | 検証: 位置シグナル強度のわずかな差異に左右されないか . . . . . | 58        |
| 4.5.1    | 方法 . . . . .                          | 59        |
| 4.5.2    | 結果 . . . . .                          | 59        |
| 4.5.3    | 考察および議論 . . . . .                     | 59        |
| 4.6      | 結論 . . . . .                          | 61        |
| 4.7      | 今後の課題 . . . . .                       | 61        |
| <b>5</b> | <b>PC を利用した並列システム: mugyu</b>          | <b>63</b> |
| 5.1      | はじめに . . . . .                        | 63        |
| 5.2      | 目的 . . . . .                          | 64        |
| 5.3      | システム構成の概要 . . . . .                   | 64        |
| 5.4      | システムのインターフェース . . . . .               | 65        |
| 5.4.1    | システムの準備時 . . . . .                    | 65        |
| 5.4.2    | システムの利用 . . . . .                     | 66        |
| 5.4.3    | ウェブインターフェース . . . . .                 | 69        |
| 5.5      | システムの仕組み . . . . .                    | 75        |
| 5.5.1    | ブート/ルートフロッピーの構成 . . . . .             | 75        |
| 5.5.2    | クライアントの設定時の GUI . . . . .             | 76        |
| 5.5.3    | クライアントの起動時に行われること . . . . .           | 76        |
| 5.5.4    | 各コマンドの仕組み . . . . .                   | 77        |

|             |                                 |            |
|-------------|---------------------------------|------------|
| 5.5.5       | ウェブインターフェース . . . . .           | 80         |
| <b>参考文献</b> |                                 | <b>86</b>  |
| <b>A</b>    | <b>実験 1</b>                     | <b>87</b>  |
| A.1         | 実験 1 で得られた最良個体 . . . . .        | 87         |
| A.2         | 最良個体の遺伝子調節タンパク質の分布の推移 . . . . . | 88         |
| <b>B</b>    | <b>実験 2</b>                     | <b>95</b>  |
| B.1         | 実験 2 で得られた最良個体 . . . . .        | 95         |
| B.2         | 最良個体の遺伝子調節タンパク質の分布の推移 . . . . . | 97         |
| <b>C</b>    | <b>実験 3</b>                     | <b>104</b> |
| C.1         | 実験 3 で得られた最良個体 . . . . .        | 104        |
| C.2         | 最良個体の遺伝子調節タンパク質の分布の推移 . . . . . | 105        |
| <b>D</b>    | <b>実験 4</b>                     | <b>132</b> |
| D.1         | 実験 4 で得られた最良個体 . . . . .        | 132        |
| D.2         | 最良個体の遺伝子調節タンパク質の分布の推移 . . . . . | 133        |

# 目 次

|      |                                                                |    |
|------|----------------------------------------------------------------|----|
| 2.1  | ショウジョウバエの胚発生のおおよその流れ . . . . .                                 | 11 |
| 2.2  | ショウジョウバエの幼生と成虫の分節の比較 . . . . .                                 | 12 |
| 2.3  | 母系効果遺伝子の相互作用 . . . . .                                         | 15 |
| 2.4  | 母系効果遺伝子とギャップ遺伝子の相互作用の関係図 . . . . .                             | 17 |
| 2.5  | ショウジョウバエ胞胚での <i>ftz</i> 遺伝子と <i>eve</i> 遺伝子による縞模様の形成 . . . . . | 19 |
| 2.6  | 単一のシグナル濃度勾配による位置特定の例 . . . . .                                 | 20 |
| 3.1  | SIMFLY2 における遺伝的オペレータ . . . . .                                 | 24 |
| 4.1  | 模式的に表したタンパク質合成の流れ . . . . .                                    | 28 |
| 4.2  | if-then ルールの集合の例 . . . . .                                     | 28 |
| 4.3  | 初期胚の遺伝子調節タンパク質の分布 . . . . .                                    | 29 |
| 4.4  | 正解の遺伝子調節タンパク質の分布 . . . . .                                     | 30 |
| 4.5  | 実験 1 の正解の遺伝子調節タンパク質の分布 . . . . .                               | 35 |
| 4.6  | 実験 1 で得られた個体の評価の過程で正解の分布にもっとも近かったとき . . . . .                  | 36 |
| 4.7  | 実験 1 における世代毎の最良個体の適応度の推移 . . . . .                             | 37 |
| 4.8  | 実験 2 における世代毎の最良個体の適応度の推移 . . . . .                             | 40 |
| 4.9  | 実験 2 の最良個体の生成した、最も正解の分布に近いステップの時の分布 . . . . .                  | 41 |
| 4.10 | <i>even-skipped</i> 遺伝子の縞形成についての仮説 . . . . .                   | 42 |
| 4.11 | 実験 3 における目指す遺伝子調節タンパク質の分布 . . . . .                            | 43 |
| 4.12 | 実験 3 における世代毎の最良個体の適応度の推移 . . . . .                             | 45 |
| 4.13 | 実験 3 における最良個体の最良ステップ . . . . .                                 | 46 |
| 4.14 | 実験 3 における最良個体の最良ステップ: 正解の分布と重ねたもの . . . . .                    | 46 |
| 4.15 | 実験 3 における最良個体の最良ステップ: <i>Krüppel</i> について . . . . .            | 47 |

|      |                                           |    |
|------|-------------------------------------------|----|
| 4.16 | 実験3における最良個体の最良ステップ: <i>giant</i> について     | 47 |
| 4.17 | 実験3における最良個体の最良ステップ: <i>hunchback</i> について | 48 |
| 4.18 | 実験3における最良個体の最良ステップ: <i>knirps</i> について    | 48 |
| 4.19 | 実験3における最良個体の最良ステップ: <i>eve</i> について       | 49 |
| 4.20 | 実験4における世代毎の最良個体の適応度の推移                    | 53 |
| 4.21 | 実験4における最良個体の最良ステップ: 正解の分布と重ねたもの           | 54 |
| 4.22 | 世代毎の最良個体の適応度の推移 (実験4とその追加実験)              | 55 |
|      |                                           |    |
| 5.1  | mugyu システムのクライアントとサーバ                     | 65 |
| 5.2  | ウェブページによる mugyu システムへのログイン画面              | 69 |
| 5.3  | mugyu システムのトップページ画面                       | 70 |
| 5.4  | mugyu システムのデスクトップページ画面                    | 73 |
| 5.5  | ウェブインターフェースを利用したジョブの投入                    | 74 |
| 5.6  | ウェブから投入したジョブの一覧を見る                        | 74 |
| 5.7  | 投入したジョブをウェブインターフェースを利用して強制終了する            | 75 |
| 5.8  | bsub から mugyud へのリクエストの送出                 | 78 |
|      |                                           |    |
| A.1  | 実験1の「拡散-合成-削除」ステップ0の各遺伝子調節タンパク質の分布        | 89 |
| A.2  | 実験1の「拡散-合成-削除」ステップ10の各遺伝子調節タンパク質の分布       | 89 |
| A.3  | 実験1の「拡散-合成-削除」ステップ20の各遺伝子調節タンパク質の分布       | 90 |
| A.4  | 実験1の「拡散-合成-削除」ステップ30の各遺伝子調節タンパク質の分布       | 90 |
| A.5  | 実験1の「拡散-合成-削除」ステップ40の各遺伝子調節タンパク質の分布       | 91 |
| A.6  | 実験1の「拡散-合成-削除」ステップ50の各遺伝子調節タンパク質の分布       | 91 |
| A.7  | 実験1の「拡散-合成-削除」ステップ60の各遺伝子調節タンパク質の分布       | 92 |
| A.8  | 実験1の「拡散-合成-削除」ステップ63の各遺伝子調節タンパク質の分布       | 92 |
| A.9  | 実験1の「拡散-合成-削除」ステップ66の各遺伝子調節タンパク質の分布       | 93 |
| A.10 | 実験1の「拡散-合成-削除」ステップ69の各遺伝子調節タンパク質の分布       | 93 |
| A.11 | 実験1の「拡散-合成-削除」ステップ72の各遺伝子調節タンパク質の分布       | 94 |
| A.12 | 実験1の「拡散-合成-削除」ステップ80の各遺伝子調節タンパク質の分布       | 94 |
|      |                                           |    |
| B.1  | 実験2の「拡散-合成-削除」ステップ00の各遺伝子調節タンパク質の分布       | 98 |

|      |                                     |     |
|------|-------------------------------------|-----|
| B.2  | 実験2の「拡散-合成-削除」ステップ05の各遺伝子調節タンパク質の分布 | 98  |
| B.3  | 実験2の「拡散-合成-削除」ステップ10の各遺伝子調節タンパク質の分布 | 99  |
| B.4  | 実験2の「拡散-合成-削除」ステップ15の各遺伝子調節タンパク質の分布 | 99  |
| B.5  | 実験2の「拡散-合成-削除」ステップ20の各遺伝子調節タンパク質の分布 | 100 |
| B.6  | 実験2の「拡散-合成-削除」ステップ25の各遺伝子調節タンパク質の分布 | 100 |
| B.7  | 実験2の「拡散-合成-削除」ステップ30の各遺伝子調節タンパク質の分布 | 101 |
| B.8  | 実験2の「拡散-合成-削除」ステップ35の各遺伝子調節タンパク質の分布 | 101 |
| B.9  | 実験2の「拡散-合成-削除」ステップ40の各遺伝子調節タンパク質の分布 | 102 |
| B.10 | 実験2の「拡散-合成-削除」ステップ42の各遺伝子調節タンパク質の分布 | 102 |
| B.11 | 実験2の「拡散-合成-削除」ステップ47の各遺伝子調節タンパク質の分布 | 103 |
| B.12 | 実験2の「拡散-合成-削除」ステップ52の各遺伝子調節タンパク質の分布 | 103 |



# 表 目 次

|      |                                                     |    |
|------|-----------------------------------------------------|----|
| 4.1  | 関数記号一覧 . . . . .                                    | 31 |
| 4.2  | 木に現れる終端記号一覧 . . . . .                               | 32 |
| 4.3  | 木に現れない終端記号一覧 . . . . .                              | 32 |
| 4.4  | 実験 1 のパラメータ . . . . .                               | 35 |
| 4.5  | 実験 2 のパラメータ . . . . .                               | 40 |
| 4.6  | 実験 3 のパラメータ . . . . .                               | 44 |
| 4.7  | 実験 4 のパラメータ . . . . .                               | 53 |
| 4.8  | 実験 4 で最良個体を更新した遺伝的オペレータと、それぞれの回数および<br>割合 . . . . . | 56 |
| 4.9  | 実験 4 での急激に適応度を良くした 2 個体の由来 . . . . .                | 58 |
| 4.10 | 初期分布を定数倍して評価した結果 . . . . .                          | 60 |

# 第 1 章

## はじめに

### 1.1 目的

ショウジョウバエの胚の中では多数の遺伝子調節タンパク質が位置シグナルとして働き、詳細な空間パターン、つまり将来の各体節となる部分の決定が行われる。この体節決定は、位置シグナルの強度のわずかな差異にあまり影響されないように行われる。このような頑健なパターン形成の仕組みは、進化の過程で段階的に形成されたと考えられる。本研究では進化的計算手法である遺伝的プログラミングを用いて、この進化の過程をシミュレートし、頑健な胚発生のパターン形成の仕組みを単純な規則の組み換えにより創発させることを目的とする。

### 1.2 背景

ショウジョウバエの幼虫に見られる特徴的な繰り返しパターンである体節を形成する先駆けとなる *even-skipped* 遺伝子 (*eve* 遺伝子) の発現の仕組みに注目した。*eve* 遺伝子は胚発生の初期段階で 7 本の縞模様を描くように発現する。いくつかの遺伝子の産物は受精前の産卵された段階で既に存在し、緩やかな濃度勾配を描いている。この濃度勾配を最初の位置シグナルとして、さまざまな遺伝子調節タンパク質が将棋倒し的に発現し、さまざまな分布を描く。*eve* 遺伝子はこれらの胚発生の初期に現れる遺伝子調節タンパク質の中で、最初に周期的な発現パターンを確立する。本研究では、この *eve* 遺伝子の発現パターンの創発に注目する。

上述のようなショウジョウバエの *eve* 遺伝子の発現の複雑な仕組みのすべてが、自然界において突然できあがったとは考えにくい。進化の過程で漸進的に作り上げられていったと考える方が自然である。より詳しく言えば、突然変異や交差などによって単純な仕組みの組み合わせが繰り返し起り、現在のような精巧な仕組みができあがったと考えられる。この単純な仕組みの組み合わせから複雑な結果を産み出す、すなわち創発させるというのは、遺伝的アルゴリズムでいうところの「積み木」が交差によって組み合わせられ最適解へ至ることとよく似ていると考えられる。このことから、本研究では創発の仕組みとして遺伝的プログラミング<sup>1</sup>を使用する。

### 1.3 実験と結果

遺伝的プログラミングによって *eve* 縞形成の仕組みを最適化させるため、遺伝子調節タンパク質間の活性化および抑制の仕組みを if-then ルールを用いて表現した。単純な記号の組合せで if-then ルールを構成し、そのルールの集合を一つの個体とみなして、胚の中央の4本縞を形成するように最適化した。その結果、求めた4本縞を形成するルールの集合を得ることができた。if-then ルールに用いる記号を注意深く限定すること、および初期分布を安定化することで、このルールの集合を得ることができた。得られたルール集合は、初期分布が12%から-5%変化しても4本縞を形成する、頑健ものであった。

また、本研究では次のような知見を得た。単純な記号の組合せでルールを形成し、遺伝的プログラミングを用いて最適化するだけで、実際のショウジョウバエに似た、複雑な遺伝子調節タンパク質の発現のダイナミクスをさまざま観察できることである。さらに、進化シミュレーションの特徴を活かし、急激に適応度を伸ばした個体について家系を調べ、どの遺伝的オペレータが効果的に働いたかを探ることができた。これらのことから、創発研究における遺伝的プログラミングの有効性を実証した。

---

<sup>1</sup>遺伝的プログラミングは、遺伝的アルゴリズムに構造的表現を扱わせることができるように拡張された最適化アルゴリズム。

## 1.4 実験環境の構築

本研究の一環としてパーソナルコンピュータ (PC) を利用した分散システム mugyu を構築した。これは縞形成の仕組みの進化シミュレーションが膨大な計算時間を必要としたためである。このシステムの特徴は、1. 計算を行う各クライアントが PC のハードディスクを利用せずフロッピーで起動する、2. 各クライアントはネットワークを介してサーバのハードディスクを利用し計算を行う、という点にある。これにより、クライアント PC の故障や電源 off が起きても、PC クラスタ全体の計算には全く影響をうけない頑健な計算が可能となった。実際、本研究では、授業では大規模に使われるが授業時間外は使用されないような実習室に設置されている数十台の PC を含めて 3 週間以上の大規模計算を行った。そして遺伝的プログラミングの分散並列環境として十分に実用に耐えられることを実証した。

## 1.5 本論文の構成

さて、この文書ではまず最初に第 2 章で本研究の生物学的背景について説明する。次に第 3 章で関連する研究について前述の目的の観点から検討する。そして第 4 章では、ショウジョウバエの *eve* 遺伝子の縞模様の形成の仕組みの進化シミュレーションで行われている処理方法を説明し、実験結果をあげ考察を述べる。最後に、第 5 章では膨大な計算時間を必要とする進化シミュレーションを行うにあたって作成した、PC を利用した分散システム mugyu について説明する。付録では各シミュレーションの結果の詳細を示す。

## 第 2 章

# 生物学的背景

この章では、本研究における生物学的背景について説明する。この章の内容は文献 [2]、[8]、[10]、[11]、[19]、[21]、[23]、[22]、[26]、[25] および [30] に拠った。

### 2.1 ショウジョウバエの胚発生のおおまかな流れ

ショウジョウバエの胚は受精後約 2 時間ほど、細胞分裂なしに核が分裂を繰り返し多核細胞になる。最初の 9 回の分裂で出来た多数の核のほとんどが卵の中心から表面へ移動して一層に並ぶ。これを多核性胞胚 (syncytial blastoderm) とよぶ。さらに 4 回核が分裂すると、卵の表面から内側に向かって細胞膜が形成されてそれぞれの核を包み込み、多核性胞胚は約 6000 個の細胞からなる細胞性胞胚 (cellular blastoderm) になる。

発生のもっとも初期の段階である多核性胞胚期 (syncytial blastoderm stage) では、共有する細胞質に多数の核を持つ単一の巨大細胞になっている。しかし細胞質は均質ではない。細胞質は、胚の長軸方向に沿って不均一に分布した遺伝子調節タンパク質の混合物を含んでいる。その不均一な分布によって、胚のある部分と別の部分とを区別する位置情報が与えられる。最初、多数の核はそれぞれ同一である。しかし、異なる遺伝子調節タンパク質にさらされるため、急速に異なる遺伝子を発現するようになる。

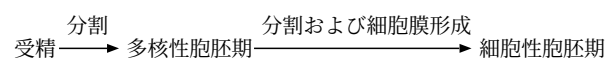


図 2.1: ショウジョウバエの胚発生のおおまかな流れ

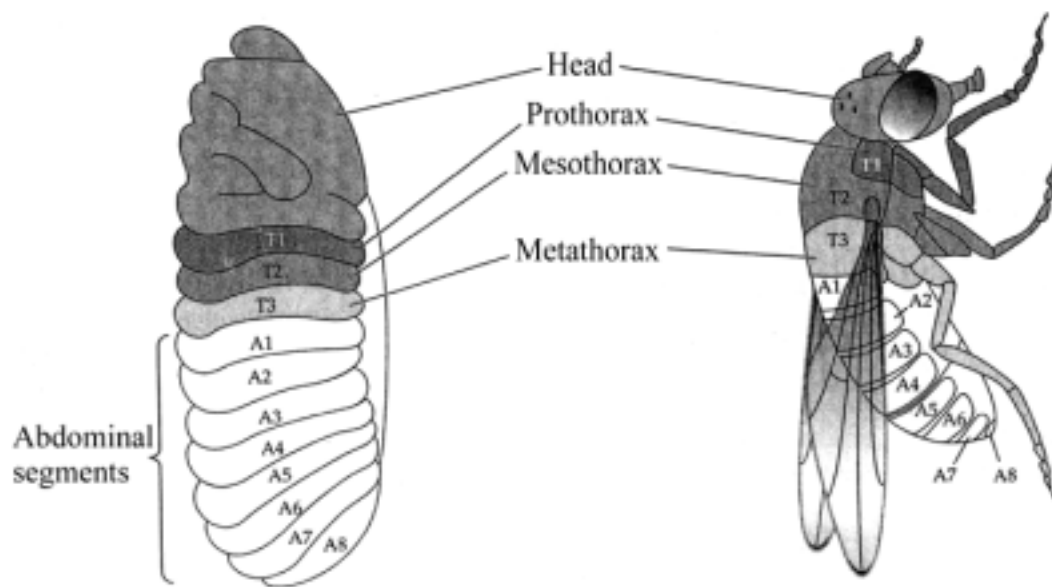


図 2.2: ショウジョウバエの幼生と成虫の分節の比較 (文献 [8] の Figure 14.2 より)。3つの分節が他と区別される。T1(prothorax) は足だけを持つ。T2(mesothorax) は翅と足を持つ。T3(metathorax) は平均体 (halter) と足を持つ。

多核性胞胚期ではそれぞれの体節を確認することはできない。しかしながら、将来体節になるそれぞれの領域における原基の分布は前述の通り、それぞれの領域で異なるものになっている。

受精後 5 ～ 8 時間で胚は胚帯伸長期に達し、原腸形成が起こり、体節が見えはじめる。10 時間目では、対軸は収縮し再び真っ直ぐになり、すべての体節が明確に区別できるようになる (図 2.2 の左図)。

## 2.2 胚の前後部軸のパターン形成モデル

ここでは、図 2.2 の左側の幼生に見られるような、胚の前後部軸の周期的な体節のパターンの形成モデルについて説明する。

### 2.2.1 母系効果遺伝子

胚、幼生、成虫の前後部極性は、卵の前後部極性に起源を持つ。ショウジョウバエの卵巣において哺育細胞によって、卵極性遺伝子の mRNA が卵母細胞に配置される。すなわち、卵極性遺伝子は卵形成の時期に母方のゲノムから転写され、その産物は受精後すぐ、ときには受精前に働きはじめる。したがって胚の表現型は、胚自身の持つ父方と母方の遺伝子の組み合わせではなく、母親の持つ遺伝子 (卵の遺伝子) によって決まる。このような様式で発現する遺伝子を母系効果遺伝子 (maternal-effect gene) とよぶ。

母系効果遺伝子には *bicoid*, *nanos*, *hunchback*, *caudal* などがある。前述の通り、卵巣の哺育細胞はこれらの mRNA を卵母細胞へ配置する。*bicoid* mRNA は前部の微小管へつなぎ止められる。*nanos* mRNA は後部の皮質細胞骨格へつなぎ止められる。*hunchback* と *caudal* mRNA は卵母細胞全体を通して分散させらる。受精がきっかけとなり、それぞれの mRNA は翻訳をはじめ。前部極では *bicoid* mRNA が Bicoid タンパク質へ翻訳され、前部がもっとも高くなり後部へ向かって下っていくような勾配を形作る。後部極では *nanos* mRNA が Nanos タンパク質へ翻訳され、後部がもっとも高くなり前部へ向かって下っていくような勾配を形作る。

胚の前部領域では、Bicoid タンパク質は *caudal* mRNA へ結合し、*caudal* mRNA の翻訳を抑制する。その結果、Caudal タンパク質は細胞の後部領域でしか合成されない。逆に、Nanos タンパク質は Pumilio タンパク質と協力して *hunchback* mRNA に結合し、胚の後部領域で *hunchback* mRNA の翻訳を抑制する。また、Bicoid タンパク質は *hunchback* 遺伝子のエンハンサー領域へ結合したり、その転写を刺激したりして、胚の前部領域における Hunchback タンパク質の量を高める。

遺伝子間のこれらの相互作用を通じて、初期胚において4つのタンパク質の勾配が形成される。

- Bicoid タンパク質: 前部から後部への勾配
- Hunchback タンパク質: 前部から後部への勾配
- Nanos タンパク質: 後部から前部への勾配
- Caudal タンパク質: 後部から前部への勾配

そして次に、この勾配が確立されている間に、分割をしていた核において、分節遺伝子 (segmentation gene) の活性の準備が整ったことになる。

これまでの説明をまとめた図を図 2.3 へ示す。

### 2.2.2 分節遺伝子

まず最初に胚の空間的なパターンを確立するのは、母系効果遺伝子であった。次に分節遺伝子 (segmentation gene) とよばれる一群の遺伝子が、胚を細分する。分節遺伝子の変異は擬体節<sup>1</sup>の欠損という形で現れる。その欠損の仕方 (変異体の表現型) と、それがどの段階で作用するか、とによって分節遺伝子は大きく 3 群に分類される。それぞれの群は順番に発現する。

最初に作用するのが**ギャップ遺伝子** (gap gene) である。ギャップ遺伝子は、母系効果遺伝子によって活性化されたり抑制されたりする。ギャップ遺伝子の産物は胚をおおざっぱに分割する。たとえば、*Krüppel* 遺伝子は胚の真中のあたり、擬体節の 4 番目から 6 番目において発現する。*Krüppel* 遺伝子の欠如によって、胚のこれらの領域が欠損する。またギャップ遺伝子産物のタンパク質は相互作用したり、ペア・ルール遺伝子の転写を活性化したりする。

次に働く分節遺伝子は**ペア・ルール遺伝子**である。ペア・ルール遺伝子は、ギャップ遺伝子によっておおざっぱに区切られた胚を細分する。ペア・ルール遺伝子の欠如によって起こされる変異は、特徴的なことに体節が一つおきに欠損して通常の半分の数しか体節がない胚となって現れる。

最後に**セグメント・ポラリティ遺伝子**がさらに胚を細分する。この遺伝子に変異が生じると、体節の数は正常だが各体節の一部が欠失し、代りに残りの部分が鏡像になって重複した構造を持つ幼虫ができる。

次にそれぞれの遺伝子群について詳しく説明していく。

---

<sup>1</sup>胚の擬体節は、幼虫や成虫の体節に直接なるものではない。擬体節の前の一部分は体節の後ろの一部分と重なり、擬体節の後ろの一部分は体節の前の一部分と重なる。



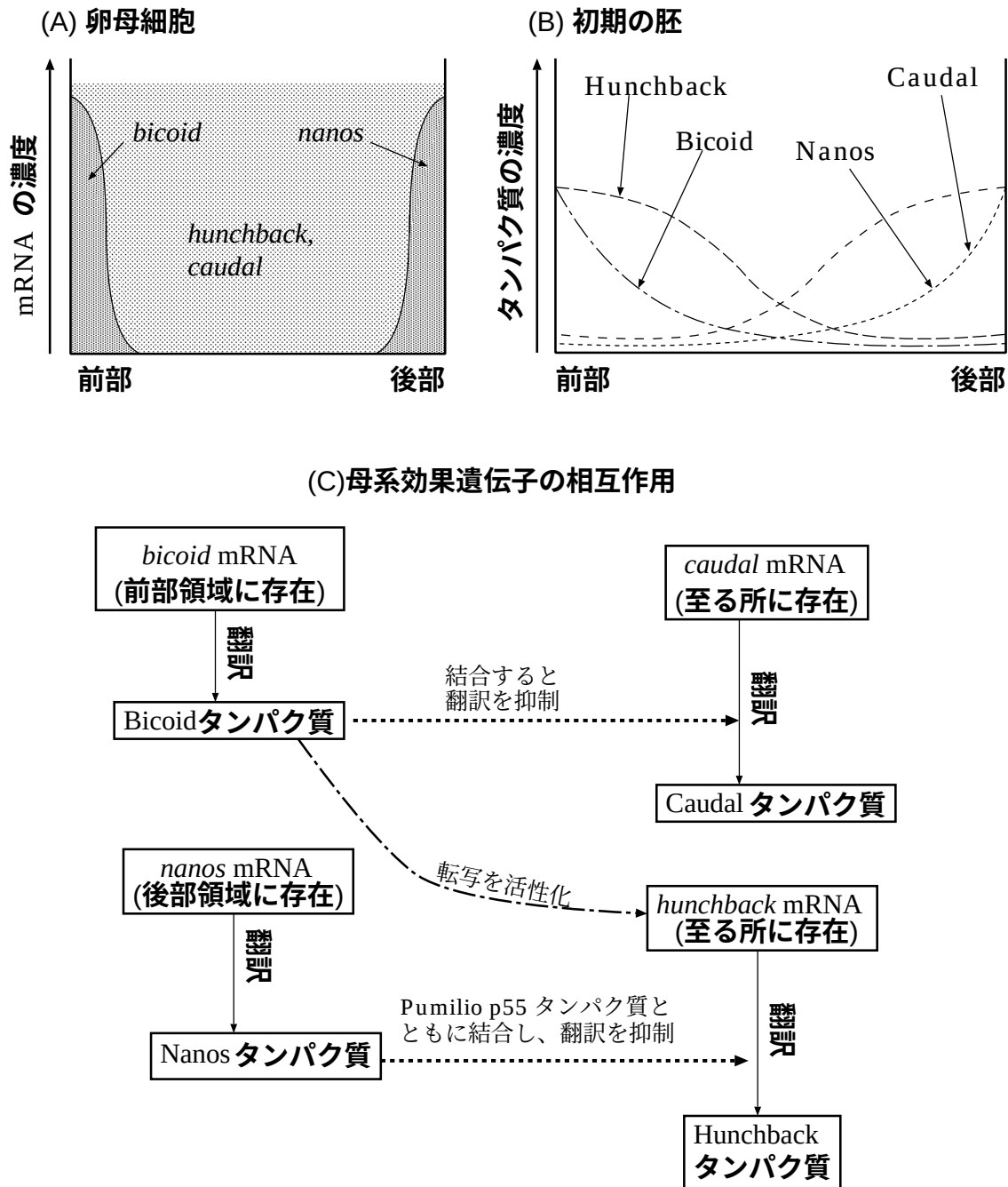


図 2.3: 母系効果遺伝子の相互作用 (文献 [8] の Figure 14.7 を参考にした)。(A) は、卵母細胞が卵巣のなかで哺育細胞に母系効果遺伝子の mRNA を設置された様子をあらわしている。(B) は、初期胚の母系効果遺伝子の産物の分布の様子を示している。(C) は、*bicoid*、*nanos*、*hunchback* および *caudal* 遺伝子の間の関係を示している。

## ギャップ遺伝子

ギャップ遺伝子は、その欠損によって体節のひと続きが失われるというように特徴づけられる。ギャップ遺伝子のなかでも特に重要なのは *hunchback* 遺伝子である。*hunchback* 遺伝子の産物である Hunchback タンパク質は、胚の前部領域で高い濃度で存在し、そのあと約 15 個の核の幅で急勾配を描いて低濃度に至り、後ろの 1/3 では Hunchback タンパク質を検出することができなくなる。母系効果遺伝子の節でも説明したとおり Bicoid タンパク質が前部領域で濃度勾配を描いている。その Bicoid タンパク質とともに Hunchback タンパク質は、前部領域のギャップ遺伝子の転写パターンを初期化する。高い濃度の Hunchback タンパク質は *giant* 遺伝子の発現を活性化する。Hunchback タンパク質が減少している領域では *Krüppel* 遺伝子の転写が起る。また、高い濃度の Hunchback タンパク質は後部領域で発現するギャップ遺伝子 (たとえば *knirps* 遺伝子) の転写を前部領域で発現しないように妨げる。

前述の通り後部領域では Hunchback タンパク質は低濃度かあるいは検出不可能なほどである。母系効果遺伝子の節で触れた Caudal タンパク質が、後部の極で最高の濃度になるような勾配を描いているが、この勾配が腹部のギャップ遺伝子である *knirps* や *giant* の活性化をしていると考えられている。*giant* 遺伝子は、前部と後部の二カ所で活性化される場所を持っている。

母系効果遺伝子と *hunchback* 遺伝子によって上述のいくつかの遺伝子の発現位置が初期化される。そのあと、ギャップ遺伝子間の相互作用によって発現が安定化および保全される。たとえば *Krüppel* 遺伝子が発現する領域の前部の境界は、Hunchback タンパク質が発現を抑制することで調節される。*Krüppel* 遺伝子発現の後部の境界は *Knirps* と *Tailless* タンパク質が発現を抑制することで調節される。Hunchback 活性がない変異体では *Krüppel* 遺伝子の発現が前部領域に向かって広がってしまい、*knirps* 活性がない変異体では *Krüppel* 遺伝子の発現が後部領域に向かって広がってしまうことが確認されている。Giant と Hunchback タンパク質が *Krüppel* 遺伝子の転写の前部の境界を調節しているように、*Krüppel* は *giant* と *hunchback* 遺伝子の転写の後部の境界を決定している。*Krüppel* 遺伝子が欠損した胚では、*hunchback* 遺伝子の転写が、通常 *Krüppel* 遺伝子が発現する場所まで広がってしまう。これらの境界を形成する抑制は、ギャップ遺伝子産物が直接お互いのコード領域に結合することで行われると考えられている。

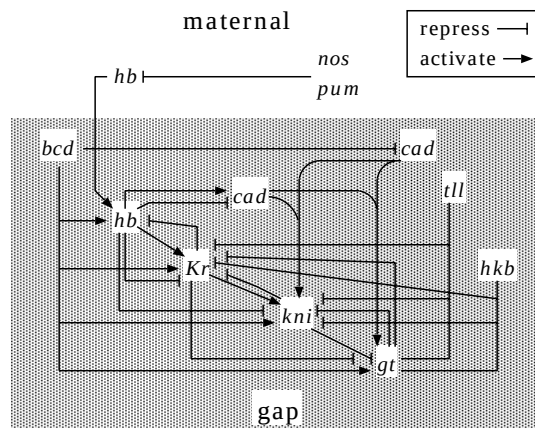


図 2.4: 母系効果遺伝子とギャップ遺伝子の相互作用の関係図。母系効果遺伝子の産物 Bicoid, Caudal, Hunchback はギャップ遺伝子の転写を調節する。Hunchback および Caudal タンパク質は母由来の mRNA によって作られる。このことから母系効果遺伝子と呼ばれるが、ギャップ遺伝子が発現する段階でも新たに転写される。そのため Hunchback および Caudal 遺伝子は図の maternal と gap の部分の両方に示されている。

以下に図中で使われている遺伝子名の略語の対応を示す。bcd → bicoid、nos → nanos、pum → pumilio、hb → hunchback、cad → caudal、Kr → Krüppel、kni → knirps、gt → giant、hkb → huckebein、tll → tailless。

また、それぞれのギャップ遺伝子の mRNA は不連続に分布している。それぞれの発現領域は隣接するかわずかに重なっているだけである。しかしながらその遺伝子産物であるタンパク質の分布は幅広く延びている。実際、すくなくとも 8 から 10 個の核の幅ほど重なっている。

## ペア・ルール遺伝子

ペア・ルール遺伝子によって、ショウジョウバエの胚に体節の最初の兆しを示される。ペア・ルール遺伝子は受精後 2、3 時間以内に活性化される。その産物である mRNA のパターンは最初は最終的な分布におおまかにしか似ていない。しかし短時間のうちに調節され、あいまいだった遺伝子産物の分布が規則正しいくっきりとした縞模様になる(図 2.5 参照)。

ペア・ルール遺伝子の分布は不安定な一時的なもので、胚発生が原腸形成から先に進むと、ペア・ルール遺伝子の産物の規則正しい分節パターンは消失する。けれども、こ

これらの作用によって胞胚の細胞に永続的な標識 (位置価) が植え付けられる。これらの位置標識はセグメント・ポラリティ遺伝子とホメオチック遺伝子の永続的な活性化として記録され、この働きで幼虫や成虫が体節構造を維持できるようになる。

図 2.5 に見られるように、ペア・ルール遺伝子は胚の前後部軸に対し垂直の縞模様を形成し、ペア・ルール遺伝子が発現しているところと発現していないところが交互に現れる。この前後部軸に沿った縞模様は胚を 15 のサブユニットに分割する。すべての核が同じペア・ルール遺伝子を発現するわけではなく、擬体節うちそれぞれの核の列で異なるペア・ルール遺伝子の組み合わせが発現していると考えられている。

3 つの遺伝子 (*hairy*, *even-skipped*, *runt*) がプライマリ・ペア・ルール遺伝子と呼ばれる。これらの遺伝子は、ギャップ遺伝子によって直接調節される。ギャップ遺伝子はプライマリ・ペア・ルール遺伝子のプロモータ領域を認識することができる。ギャップ遺伝子産物の異なる濃度の組み合わせによって、遺伝子が転写されるかどうか決定されることが考えられている。

この遺伝子調節タンパク質の濃度の異なる組み合わせに反応するという遺伝子スイッチの仕組みはモジュール的である。たとえば *even-skipped*(*eve*) 遺伝子の転写調節領域は極めて大きく (2 万塩基対)、一連の比較的単純な調節モジュールから出来ている。それぞれのモジュールは複数の調節配列を含み、*eve* 遺伝子の特定の縞を決めるのに関わっている。*eve* 遺伝子の 2 番目の縞の調節モジュールを例にあげると、このモジュールには *eve* の転写を活性化する二つの遺伝子調節タンパク質 (Bicoid と Hunchback) と転写を抑制する二つの遺伝子調節タンパク質 (Krüppel と Giant) の識別コードが含まれている。これらの 4 つのタンパク質の相対濃度により、*eve* 遺伝子の転写をオンにするタンパク質複合体が縞 2 モジュールで出来るかどうかが決まる。このような仕組みで胚の唯一の場所にしか *eve* の縞 2 が発現しないようになっている。

ギャップタンパク質によって初期化されると、プライマリ・ペア・ルール遺伝子の転写パターンは、それらプライマリ・ペア・ルール遺伝子自身の相互作用によって安定化される。さらに、プライマリ・ペア・ルール遺伝子はそれに続くセカンダリ・ペア・ルール遺伝子 (*fushi tarazu* 遺伝子など) の発現を調節する。14 番目の核分裂期では *fushi tarazu*(*ftz*) RNA とそのタンパク質は胚全体を通して分布している。しかし、プライマリ・ペア・ルール遺伝子産物が *ftz* 遺伝子のプロモータと相互作用しはじめると、*ftz* 遺伝子の縞模

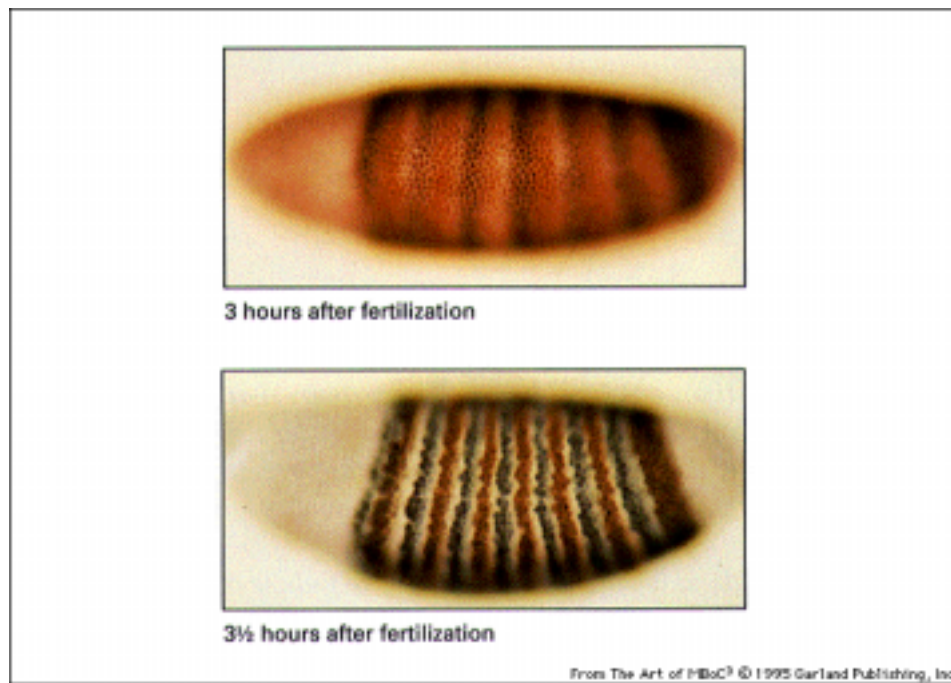


図 2.5: ショウジョウバエ胚での *ftz* 遺伝子と *eve* 遺伝子による縞模様の形成。*ftz* と *eve* はともにペア・ルール遺伝子で、その発現パターン (*ftz* は茶色で、*eve* は灰色で示されている) は最初はボンヤリしているが、すぐにくっきりした縞模様になる。(文献 [2] の Fig. 21-65 より)

様の中の部分で発現が抑制され、*ftz* 遺伝子の縞がくっきりとしはじめる。一方で *ftz* 遺伝子産物は *ftz* 遺伝子自身のプロモータと相互作用をはじめ、*ftz* 遺伝子の転写をより刺激する。

### セグメント・ポラリティ遺伝子

セグメント・ポラリティ遺伝子は、擬体節ごとに繰り返すパターンで発現する。たとえば *engrailed* 遺伝子は細胞性胚期において、その転写産物 RNA を 14 本の縞模様に発現する。この縞はどれもほぼ細胞 1 個の幅で、その位置は将来の擬体節の最前方に相当する。この縞はペア・ルール遺伝子の発現の縞と一定の位置関係にある。セグメント・ポラリティ遺伝子も、その発現パターンは階層構造で一段階上位にある遺伝子産物の組み合わせによって制御され、セグメント・ポラリティ遺伝子間の相互作用によってさらに細かく精錬される。

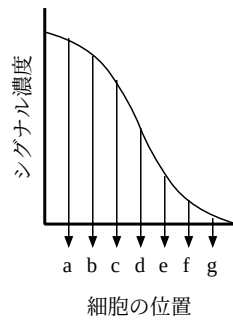


図 2.6: 単一のシグナル濃度勾配による位置特定の例。一つの位置シグナルの濃度のわずかな違いにより、細胞の位置 a から g を特定する。

ペア・ルール遺伝子までの遺伝子間の相互作用は、多核性胞胚の中での分子間の相互作用であった。細胞膜が核を覆い、細胞性胞胚期に入ると細胞間の相互作用によって形態形成が行われるようになる。細胞間の相互作用はセグメント・ポラリティ遺伝子によって行われる。セグメント・ポラリティ遺伝子の多くは、細胞のシグナル伝達の構成要素であるタンパク質をコードしている。セグメント・ポラリティ遺伝子は細胞間でシグナルを送りあって、それぞれの擬体節の細胞の運命を決定づける。

## 2.3 まとめ

ショウジョウバエの胚発生において、細胞を特定する仕組みは次のようなものであった。まず母系効果遺伝子 (*bicoid* や *nanos* など) の産物が胚全体におよぶ緩やかな濃度勾配を形成する。そしてその濃度勾配が胚全体における位置シグナルとなる。次に、ギャップ遺伝子 (*Krüppel* や *giant* など) のそれぞれが母系効果遺伝子による濃度勾配に反応し、特定の領域でそれぞれ発現する。そして、ギャップ遺伝子の産物の不均一で局所的な分布がその次の段階の位置シグナルとして働き、周期的なパターンを描くペア・ルール遺伝子 (*even-skipped* など) の発現を調節する。ペア・ルール遺伝子はセグメント・ポラリティ遺伝子などの遺伝子の発現に影響をおよぼして、さらに詳細なパターンを調節する。このようにして、母系効果遺伝子が産み出す胚全体にわたる濃度勾配が、位置調節の階層構造の最初の階層になり、胚を順次細分化し精密なパターンを作り出すものになる。

上のような細胞を特定する仕組みとは別な仕組みを考えることができる。その仕組み

とは、一つの位置シグナルの濃度勾配しかなく、各細胞が位置シグナルの濃度のわずかな違いに正確に反応しなくてはならないという仕組みである (図 2.6 参照)。この仕組みの場合、わずかな濃度の違いに反応しなくてはならないので、位置シグナルの設定のわずかな誤りで細胞の特定に失敗してしまう。実際の細胞を特定する仕組みは、複数の位置シグナルを段階的に設定するため、より信頼性の高い仕組みとなっている。

## 第 3 章

# 関連する研究

ここでは、主に関連する研究で用いられている「縞模様の形成の仕組みの進化シミュレーション」の手法についてとりあげ、続く節で議論する。

### 3.1 SIMFLY および SIMFLY2

SIMFLY[4] および SIMFLY2[3] は、Arita らにより開発されたシステムで、胚発生におけるパターン形成のルールを確認、訂正する。SIMFLY システムにおけるシミュレータはすべての可能なルールのパターンを調べ、評価プログラムはシミュレータの結果を量的に評価する。また、次の点について焦点を当てている。

1. DNA 結合タンパク質が本当にスイッチとして働くのか
2. 現在示されている知識は、分節化の仕組みを説明するのに十分かどうか
3. 同じ結果を生む他のモデルの探索

SIMFLY システムはルールベースのシミュレータで、キイロショウジョウバエ (*Drosophila melanogaster*) の多核性胞胚期から細胞性胞胚期に至る過程の分節化をシミュレートする。胚のモデルは、細胞の帯の輪を表現する「カラム」のリストからなりたっている。すなわちカラムは、楕円形をした胚の輪切りの一枚一枚を表現する。DNA 結合タンパク質の遺伝子スイッチとしての機能は if-then ルールとして表現される。二つのタンパク質質間の関係は抑制する関係、または活性化する関係として記述され、システムはその閾値を探索する。



シミュレータはそれぞれの時刻で、タンパク質を削除し、新しいタンパク質を合成し、タンパク質を拡散する。これらの処理を一定の時間繰り返す。それぞれのルールについて、シミュレータはタンパク質の分布を形作ることになる。

Arita らは SIMFLY システムにおいてルール探索を次のようにして行った。固定された母系効果タンパク質 *bicoid* と *nanos* の勾配から、ギャップタンパク質の勾配を作るルールを探した。if-then ルールの条件部は「 $bic > 3.0 \ \& \ nos < 1.0$ 」のようになる。そのうち閾値の部分(前の例では 3.0 と 1.0 にあたる)に 1, 3, 5 のどれかをあてはめ、網羅的に探索を行った。

Arita らは SIMFLY2 では遺伝的アルゴリズム (GA) を採用した。GA を適用するにあたり、次のように if-then ルールを形式化した。

まずタンパク質間の関係の種類を以下のように分類した。

S: Squelching. つづく二つの整数は閾値の上限と下限になる。下限の閾値が上限の閾値より大きい場合は、タンパク質は相互作用しない。

A: Activation. 続く整数は閾値となる。閾値の上限は 5 である。

R: Repression. 続く整数は閾値となる。閾値の下限は 0 である。

N: No relation. タンパク質は相互作用しない。

if-then ルールの条件部を次のように形式化した。

```
bcd    caudal  hb      Kr      kni    giant
[(A 2) (N .)  (N .) (R 3) (N .) (N .)]
```

上の例は、「*bicoid* が 2 以上あり、*Krüppel* が 3 以下であれば」と解釈する。また、次の例 “ $[(S1) (S2)] [(S3) (S4)]$ ” は OR で条件がつながっていることを示し、“ $(S1 \wedge S2) \vee (S3 \wedge S4)$ ” を意味する。

次に完全な if-then ルールの例を示す。

```
[hairy 10 1 [
[(N .) (N .)  (N .) (N .)    (S 5 4) (N .)  (N .) (N .)]
[(N .) (N .)  (N .) (N .)    (N .)    (S 3 1) (N .) (N .)]
```

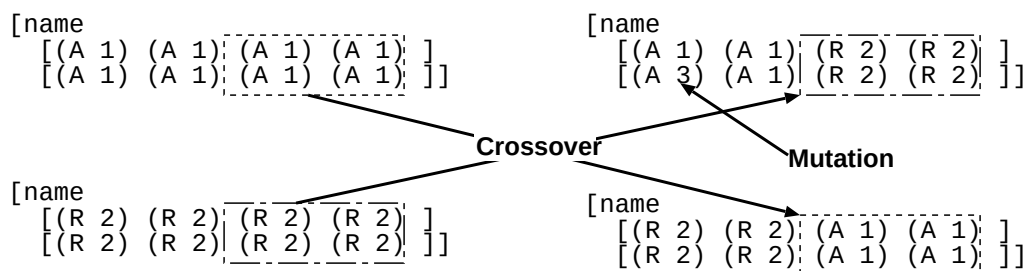


図 3.1: SIMFLY2 における遺伝的オペレータ。交差と突然変異の例

```
[(N .) (N .) (N .) (S 4 2) (N .) (N .) (N .) (N .)]]]
bcd   caudal  hb    Kr      kni    giant  hairy  runt
```

上のルールのうち hairy はタンパク質名を示し、条件部が満たされると hairy タンパク質が合成されることになる。タンパク質名に続く 10 1 は拡散定数を示す。この値はシミュレーション中は変化しない。

図 3.1 に、SIMFLY2 における遺伝的オペレータ (交差と突然変異) がどのようにルールに作用するかを示す。

## 3.2 “Simulation of *Drosophila* Embryogenesis”

Hamahashi らは文献 [9] で、ショウジョウバエの初期の胚発生の詳細なモデルを作成した。その特徴は、遺伝子発現パターンの時間的なダイナミクスを観察することを可能としたことである。詳細なモデルを用いたシミュレーションは、実際のショウジョウバエの遺伝子発現パターンを再現した。とくに、機能喪失ノックアウト実験<sup>1</sup>のシミュレーションについても、実際の生物学的な機能喪失ノックアウト実験と同じような遺伝子発現パターンを示した。

Hamahashi らの基本的な方針は、次のようなものであった。

- 生物学的実験により知られているメカニズムを実装し、初期の胚発生をできるだけ正確におこなえるシミュレータを作成する。

<sup>1</sup>目標となる遺伝子の転写を不可能にする実験

- その既知のメカニズムを用いて、シミュレータ上でショウジョウバエの変異体を観察し、
- よく知られている生物学的成果に隠れている、現在はまだよく知られていないメカニズムを予測する。

実験は二つのステップからなる。

1. シミュレーションに使われる多くのパラメータ (タンパク質の拡散定数、DNA への結合親和性、転写率など) を、遺伝的アルゴリズムを用いて野生型の胚の遺伝子発現パターンに近くなるように最適化する。
2. 1 番目のステップで最適化されたパラメータ集合を調節することで、変異体をシミュレートする。

### 3.3 関連する研究についての議論

同じ結果を生む他のモデルの探索を主な目的とするという観点から見ると、Arita らの SIMFLY[4] および SIMFLY2[3] のルール形式は柔軟でない。とくに SIMFLY では、ルールの条件部のタンパク質名は固定され、閾値のみを網羅的に探索した。SIMFLY2 では、条件部を遺伝的アルゴリズムで交差や突然変異でかき混ぜたが、あるタンパク質を合成する新たなルールを自由に (突然変異などで) 生成する仕組みはなかった。

Hamahashi らはショウジョウバエの初期の胚発生の詳細なシミュレーションを行ったが[9]、詳細な空間パターンを形成する仕組みを創発させるという観点には立っていない。

## 第 4 章

# 縞形成の仕組みの進化シミュレーション

### 4.1 はじめに

縞形成の仕組みの進化シミュレーションをするプログラムは、ショウジョウバエの初期胚の状態から *eve* の縞を発現させるタンパク質合成の規則の集合を、進化的計算手法である遺伝的プログラミング<sup>1</sup> (GP: Genetic Programming) を用いて探索する。この章では、縞形成の仕組みの進化シミュレーションの目的、方法、実験結果、考察について述べる。

### 4.2 目的

ショウジョウバエの胚の中では多数の遺伝子調節タンパク質が位置シグナルとして働き、詳細な空間パターン、つまり将来の各体節となる部分の決定が行われる。位置シグナルの強度のわずかな差異にあまり影響されないように、この過程は段階的に行われる。進化の過程で、位置シグナル強度のわずかな差異に左右されない段階的なパターン形成の仕組みが得られたことが予想される。本研究では進化的計算手法を用い、この進化の過程をシミュレートし、詳細な空間パターンを形成する仕組みを創発させることを目的とする。

---

<sup>1</sup>遺伝的アルゴリズムおよび遺伝的プログラミングについての詳細は、参考文献 [1]、[12]、[13]、[14]、[15]、[16] および [17] を参照のこと。

## 4.3 方法

### 4.3.1 プログラムで使用されているモデル

計算機上で進化的計算方法を用いて縞模様を発現させる仕組みを創発させるため、ショウジョウバエの胚をできるだけ簡略化した。モデルでの初期の胚には、ショウジョウバエの卵と同じように卵極性遺伝子の産物による遺伝子調節タンパク質の勾配を設定しておく (図 4.3)。

また、モデルでの初期胚は、受精から細胞性胞胚期に至るまでの過程のうち多核性細胞期に似た状態<sup>2</sup>を想定しモデル化したものである。

実際の胚発生の核はモルフォゲン勾配にさらされながら核分裂を繰り返し、細胞性胞胚期に至る [2]。しかし、進化的シミュレーションを段階的なパターン形成の仕組みを探索・創発することに集中させたいため、核分裂の過程をシミュレーションのモデルに採り入れることは避け、核の個数は固定した。

遺伝子調節タンパク質の合成、拡散および削除が行われる場合は、次のようにデータを構造化した。

- それぞれの遺伝子調節タンパク質の濃度情報が納められている**細胞**
- その細胞の一次元配列である**胚**

実際のショウジョウバエの胚は紡錘形をしており、多核性胞胚期ではその表面上におよそ 4000 個の核が存在している [2]。プログラムで使用されているモデルでは、タンパク質合成メカニズムは単純化されている。実際のタンパク質の合成は、図 4.1 の a) のように行われるが、モデルではそれを単純化し図 4.1 の b) のように行う。

遺伝子調節タンパク質がある遺伝子を活性化したり抑制することは、if-then ルールのように書ける。したがって DNA は if-then ルールの集合とみなすことができる。図 4.2 は、if-then ルールの集合の例である。if-then ルールの集合を、遺伝的プログラミングにおいて 1 個体として扱い最適化する。

---

<sup>2</sup>多核細胞から核が胚の周辺に移動し、細胞の境界形成が始まる直前の状態

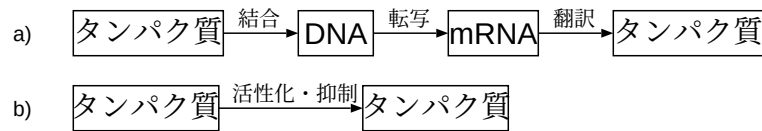


図 4.1: 模式的に表したタンパク質合成の流れ。実際は図の上の方の a) に近いが、モデルでは b) のようにタンパク質の合成が行われる。

```

(IF (> (ConcentrationOf protein9) 30.611979) protein9)
(IF (< (ConcentrationOf protein3) 13.232692) protein1)
(IF (< (ConcentrationOf protein2) 1.176075) protein1)
(IF (> (ConcentrationOf protein7) 5.285993) protein5)
  
```

図 4.2: if-then ルールの集合の例。例えば、1 行目は「もしタンパク質 9 の濃度が 30.611979 より高ければ、タンパク質 9 を 1 単位合成せよ」と解釈する。

### 4.3.2 個体の評価方法

個体の評価方法の大まかな流れは以下のようなになる。

1. **胚の初期化:** 胚の中の遺伝子調節タンパク質の分布を、あらかじめ決められた分布 (図 4.3) をとるように初期化する。
2. 以下、「**合成-拡散-削除**」ステップと呼ぶ処理を、500 ステップから 400 ステップ (実験によって異なる) 繰り返す。
  - (a) **タンパク質の合成:** 胚の各細胞において、if-then ルールの集合が評価され、タンパク質が合成される。各細胞ごとに遺伝子調節タンパク質の濃度の組合せが異なるので、細胞の各々で遺伝子調節タンパク質が異なった量だけ合成される。ある細胞であるタンパク質が合成されると、その細胞内のそのタンパク質の濃度の量が 1 単位増えることになる。
  - (b) **タンパク質の拡散:** 前述の通り、胚は細胞の一次元配列である。各細胞は各遺伝子調節タンパク質の  $1/4$  づつを両隣の細胞に渡す。最左と最右の細胞は各遺伝子調節タンパク質の  $1/2$  を隣の細胞に渡す。
  - (c) **タンパク質の削除:** 各細胞で、各タンパク質の量が 1 単位だけ削除される。つまりタンパク質の濃度は、そのタンパク質を保全するようなルール (例えばポジティ

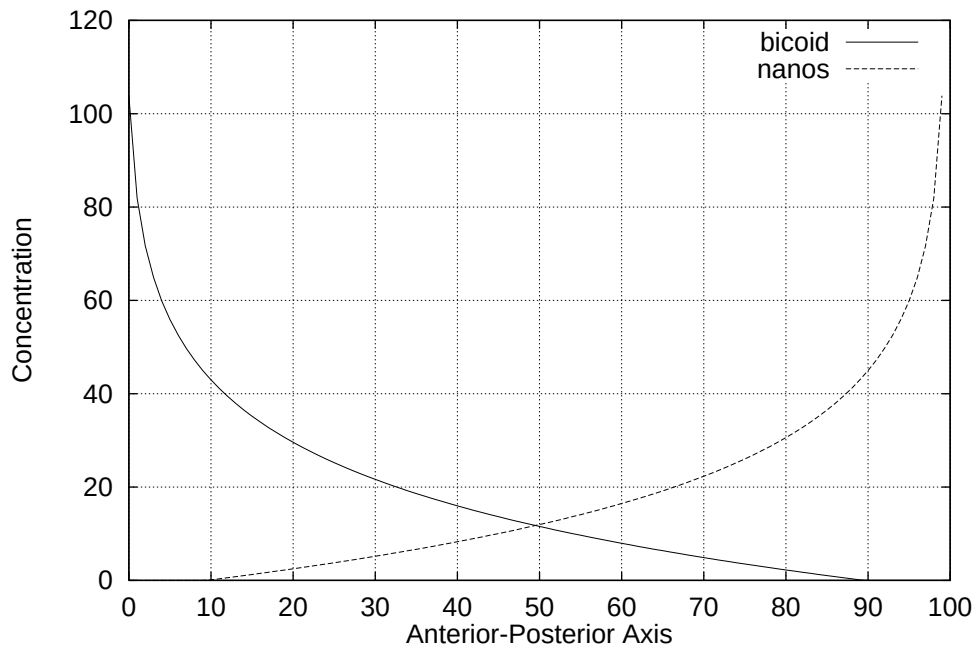


図 4.3: プログラムで使用する初期胚の遺伝子調節タンパク質の分布の例。横軸は胚の前後部軸で、縦軸はタンパク質濃度を示す。図からは 100 個の細胞において、*bicoid* タンパク質の濃度の分布が前から後ろへ緩やかな勾配を描き、*nanos* タンパク質の濃度の分布が後ろから前へ緩やかな勾配を描いている様子が見て取れる。なお、初期胚の遺伝子調節タンパク質の分布は問題ごとに異なる。

ブフィードバックがかかるようなルール) がないと、すぐに濃度は 0 になってしまう。

3. **正解の分布との比較:** あらかじめ与えられた正解の遺伝子調節タンパク質の分布 (図 4.4 など、実験によって異なる) との比較が行われる。各細胞で、正解として与えられた遺伝子調節タンパク質のそれぞれの濃度と現在のそれぞれの濃度の差を求め、足し込んだものが適応度とされる。適応度は 0 に近ければ良いことになる。

### 4.3.3 プログラムの実行の大まかな流れ

#### 1. 初期化

- 胚の初期化: 初期胚のデータおよび、正解の胚のデータの読み込み

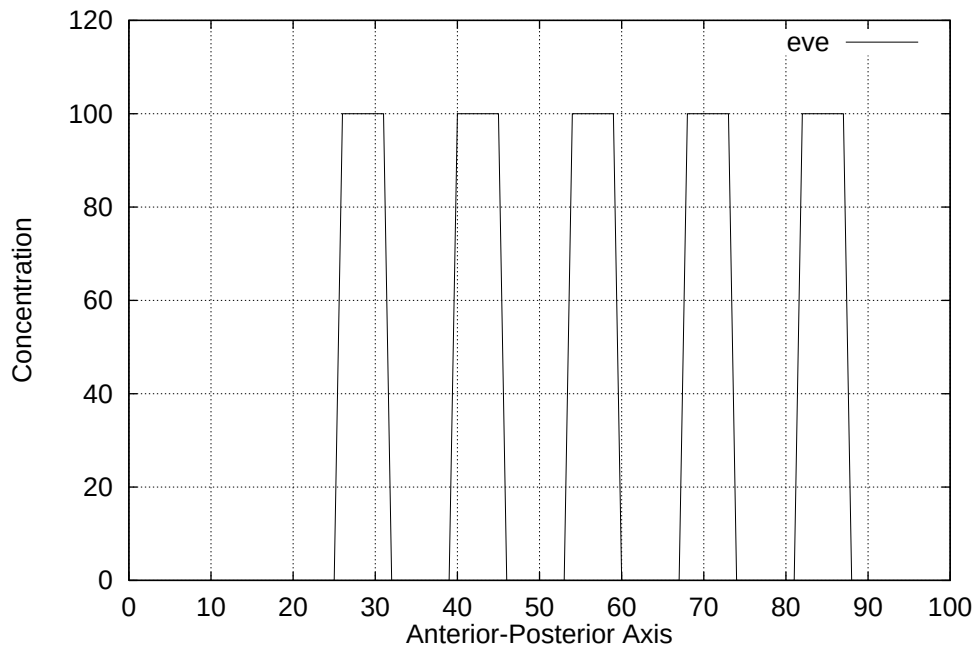


図 4.4: プログラムで使用される正解の遺伝子調節タンパク質の分布の例。実験ごとに異なる。

- 個体の集団の初期化: 所定の個体数分だけ、ランダムにルールを生成する。

2. 最適化: 以下のステップを繰り返す。

- 個体の集団 (1000 から 2000 個体。実験によって異なる) の中から所定の個体数分 (7 から 14。実験によって異なる) だけランダムに選び、トーナメント用の個体とする。
- 遺伝的オペレータの集合から、適用する遺伝的オペレータを選ぶ。(実験に使用された遺伝的オペレータについては、第 4.3.5 節を参照)
- トーナメントする (選ばれた個体たちを適応度順に並べる。)
- トーナメント順位の上の方、すなわち適応度の良い方から、選ばれた遺伝的オペレータに必要な分だけ個体を選んで遺伝的オペレータにかける。
- 遺伝的オペレータの結果得られた個体を、トーナメント順位の下の方の個体と入れ換える。



表 4.1: 関数記号一覧

| 記号名             | 印字名             | 引数の型                             | 返し値の型           |
|-----------------|-----------------|----------------------------------|-----------------|
| IF              | IF              | BooleanConstant, Protein         | ConcreteProtein |
| ConcentrationOf | ConcentrationOf | Protein                          | Concentration   |
| GraterThan      | >               | Concentration, RealNumber        | BooleanConstant |
| SmallerThan     | <               | Concentration, RealNumber        | BooleanConstant |
| AND             | AND             | BooleanConstant, BooleanConstant | BooleanConstant |
| OR              | OR              | BooleanConstant, BooleanConstant | BooleanConstant |
| NOT             | NOT             | BooleanConstant                  | BooleanConstant |

#### 4.3.4 使用している記号

ここではルールの中で使用している関数記号および終端記号について説明する。「縞形成の仕組みの進化シミュレーション」のプログラムでは、型つき GP[20] を使用している。そのため各表 (表 4.1、表 4.2、表 4.3) には記号名と印字名の他に、その記号の取る引数の型とその記号を評価すると返される値の型を加えた。表 4.1 に、関数記号一覧を示す。記号 GraterThan のように、記号の名前と印字される文字列が違う場合があるので、表では記号名と印字名の二つを挙げた。また、表中の引数の型でカンマで区切られ二つの型名がある場合は、その記号は二つの引数をとることを示す。表 4.2 に、木に現れる終端記号一覧を示す。終端記号は評価されると自分自身と同じ型の値を返すので、返し値の型の項目は省略した。また、終端記号はいわゆるプログラム中の「定数」にあたるので、異なるインスタンスでは同じ型でも値が異なることから、関数記号の表で示した「印字名」の代りに「印字例」を示していることに注意されたい。また、印字例とともに取り得る値の範囲を示してある。さらに表 4.3 に、木に現れない終端記号一覧を示す。木に現れない終端記号は、木の評価時にインスタンス化され内部的に使用される。

終端記号 RealNumber と Concentration は本質的には同じものである。この二つを分けたのは、関数記号 GraterThan (>) と SmallerThan (<) の第 1 引数が必ずあるタンパク質の濃度を指すようにし、第 2 引数が単なる実数値であるようにするためである。終端

表 4.2: 木に現れる終端記号一覧。終端記号は評価されると自分自身と同じ型の値を返す。

| 記号名        | 印字例      | 取り得る値の範囲                     |
|------------|----------|------------------------------|
| Protein    | protein1 | protein1 から protein9 (実験による) |
| RealNumber | 12.345   | 0 から 100 (単精度浮動小数点数)         |

表 4.3: 木に現れない終端記号一覧。これらの記号は木の評価中にインスタンス化される。表 4.2 の見出しでも触れた通り、終端記号は評価されると自分自身と同じ型の値を返す。

| 記号名             | 印字例            | 取り得る値の範囲                     |
|-----------------|----------------|------------------------------|
| Concentration   | 12.345         | 0 から 100 (単精度浮動小数点数)         |
| ConcreteProtein | protein1       | protein1 から protein9 (実験による) |
| BooleanConstant | true または false | 印字例と同様                       |

記号 Protein と 終端記号 ConcreteProtein についても本質的に同じものである。システム内では、ランダムにルール文を生成する際に「タンパク質を返すような文を生成せよ」と指定する。終端記号 Protein は評価するとタンパク質を返すので、これだと、IF 文を根に持つようなルールが生成されず単に 'Protein9' というような文が生成されてしまう可能性がでてくる。そのため、ルール文を生成する際には「ConcreteProtein を返すような文を生成せよ」と指定し、さらに、評価すると ConcreteProtein を返すような記号は IF だけにし、IF 文の第 2 引数に求められる型を Protein にした。

#### 4.3.5 使用している遺伝的オペレータ

ここでは実験で使用された遺伝的オペレータについて説明する。以下の各節では、1. 遺伝的オペレータに必要な親の数、2. オペレート後に返す子の数、3. オペレータの作用について説明する。なお、使用される遺伝的オペレータの集合および、それぞれの遺伝的オペレータが選択される確率は実験ごとに異なる。

**交差**                      必要な親の数は 2 である。返す子の数は 2 である。親のそれぞれからランダムに部分木を 1 つずつ選び、入れ換える。部分木がランダムに

選ばれる際に、関数記号が選ばれる確率は 0.7 である。

- 突然変異** 必要な親の数は 1 である。返す子の数は 1 である。親からランダムに部分木を 1 つ選び、ランダムに生成した部分木と取り換える。部分木がランダムに選ばれる際に、関数記号が選ばれる確率は 0.7 である。ランダムに生成する木の深さの最大は 3 である。
- 個体の複製** 必要な親の数は 1 である。返す子の数は 1 である。親を複製して返す。
- ルールの交差** 必要な親の数は 2 である。返す子の数は 2 である。二つの親からランダムにルールを選び入れ換える。
- ルールの追加** 必要な親の数は 1 である。返す子の数は 1 である。親にランダムに生成したルールを加えた子を返す。
- ルールの削除** 必要な親の数は 1 である。返す子の数は 1 である。親からランダムにルールを 1 つ選択し、それを削除した子を返す。
- ルールの複製** 必要な親の数は 1 である。返す子の数は 1 である。親からランダムにルールを 1 つ選択し、それを複製して加えた子を返す。

## 4.4 実験結果

この節では各実験の目的、方法、結果を示し、考察を述べる。実験結果の図の数が膨大なものは付録に別掲した。各実験の結果を示した後にその実験固有の事柄についての考察が書かれていることがあるが、総合的な考察については別に節をもうけてある。

### 4.4.1 実験 1

#### 目的

この実験は「縞形成の仕組みの進化シミュレーション」のシステムの評価のために行った。

## 方法

正解の遺伝子調節タンパク質の分布の形を縞模様ではなく、より単純な形である紡錘形 (図 4.5) に設定し最適化を行った。詳しく言えば、ルールのなかで “protein9” とある仮想的なタンパク質の濃度の分布を、正解の分布である紡錘形 (図 4.5) と比較する。比較の方法は、第 4.3.2 節で述べた通りである。つまり、仮想的な核のそれぞれで、正解として与えられた遺伝子調節タンパク質のそれぞれの濃度と現在のそれぞれの濃度の差を求め、足し込んだものが適応度とされる。適応度は 0 に近ければ良いことになる。実験のパラメータを表 4.4 に示す。

## 結果

実験の結果、正解と良く似た形の分布 (図 4.6 参照) を形成するルールの集合が得られた。3 回の試行での、世代毎の最良個体の適応度の推移を図 4.7 に示す。3 回の試行で見つかった最良個体を付録 A.1 に示す。付録 A.1 で示した最良個体の「合成-拡散-削除」ステップを重ねる様子を付録 A.2 に示す。

## 考察

図 4.6 を見るとわかる通り、正解と良く似た形の分布を形成するルールの集合が得られた。このことから、「縞形成の仕組みの進化シミュレーション」のシステムが、正解として与えた分布を形成するルール集合を得ようと最適化できることが実証された。

### 4.4.2 実験 2

#### 目的

この実験では、近似的な胚の初期の段階 (図 4.3) から縞模様 (図 4.4) を形成するルールの集合を得ることを目指した。

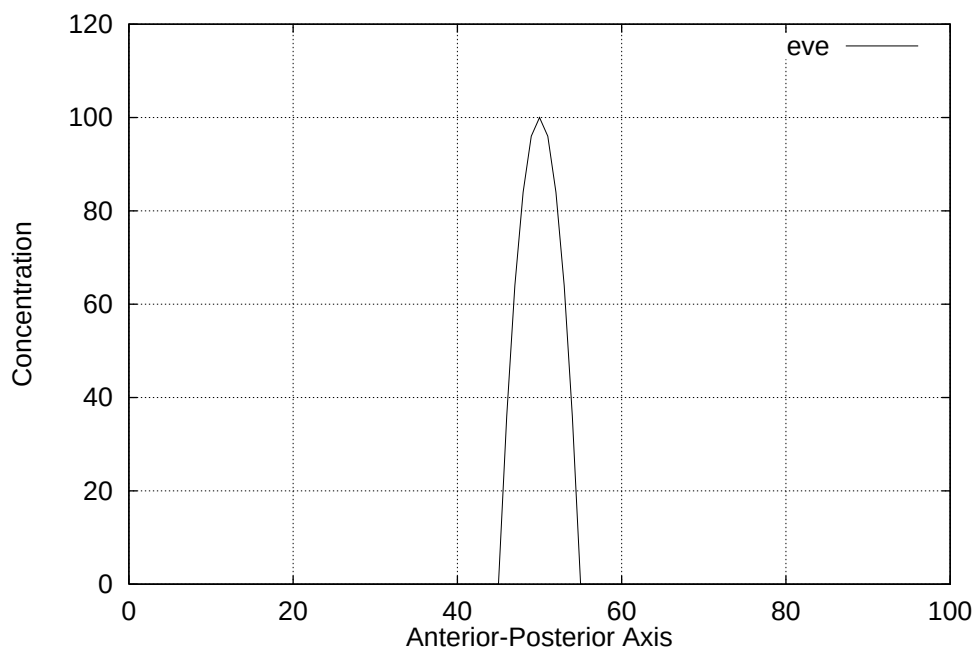


図 4.5: 実験 1 の正解の遺伝子調節タンパク質の分布

表 4.4: 実験 1 のパラメータ

| パラメータ                  | 値                       | 遺伝的オペレータ | 確率                  |
|------------------------|-------------------------|----------|---------------------|
| 個体数                    | 2000                    | 交差       | $12/30 = 0.4$       |
| 初期化時の木の深さ              | 3                       | 突然変異     | $2/30 \simeq 0.067$ |
| 初期化時のルールの数             | 20                      | 個体の複製    | $1/30 \simeq 0.033$ |
| トーナメントサイズ              | 7                       | ルールの交差   | $8/30 \simeq 0.267$ |
| 最大の「合成-拡散-削除」<br>ステップ数 | 500                     | ルールの追加   | $5/30 \simeq 0.167$ |
| 分布の比較をはじめるス<br>テップ数    | 0                       | ルールの削除   | $2/30 \simeq 0.067$ |
| タンパク質を表す記号で<br>使用されるもの | protein0 から<br>protein9 |          |                     |

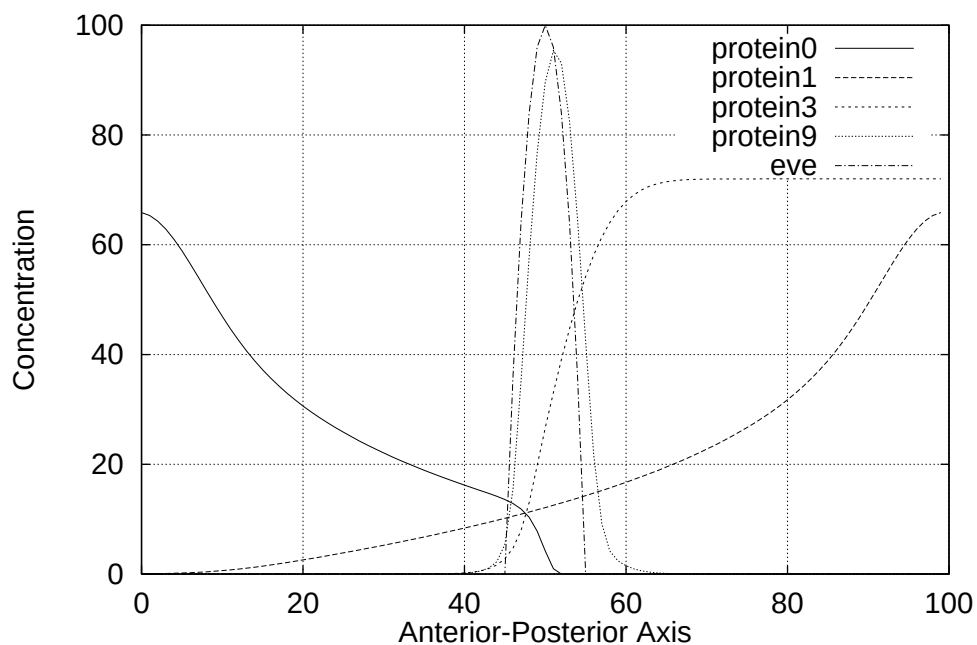


図 4.6: 図は実験 1 で得られた個体の評価の過程で正解の分布にもっとも近かったときである。この分布は第 72 「合成-拡散-削除」ステップ目で得られた。図中で “protein0” と示されているものは、実験で bicoid タンパク質に相当するものとして扱われている。同様に、“protein1” は nanos、“protein9” は eve である。図中で “eve” と示されている分布は正解の分布である。最適化によって得られたルールが生成した “protein9” の分布と、正解の “eve” の分布が良く重なっているのが見て取れる。

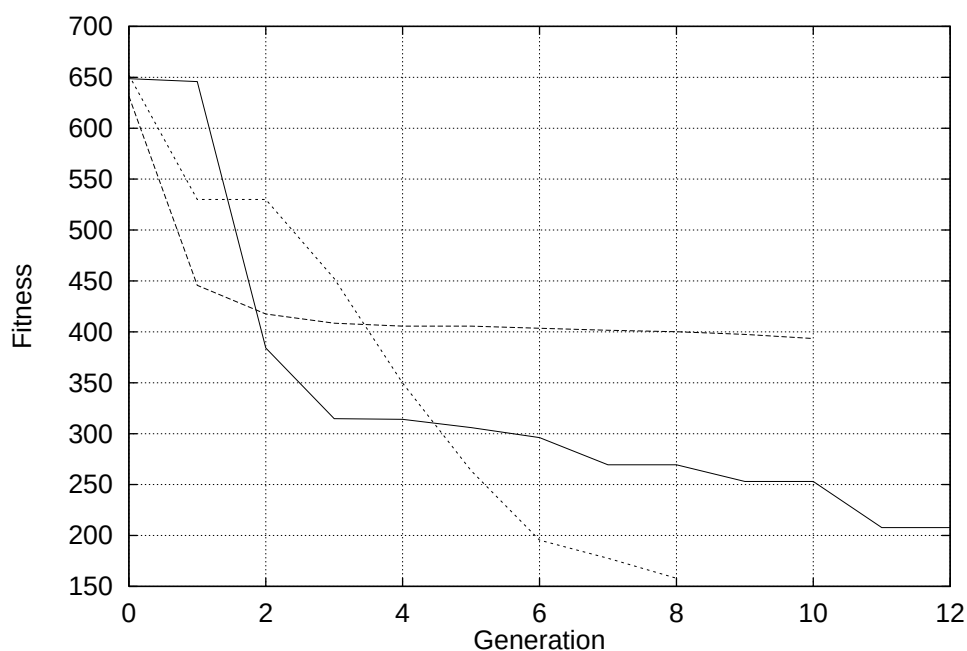


図 4.7: 実験 1 における世代毎の最良個体の適応度の推移

## 方法

実験方法は第 4.3 節で示した方法を用いている。この実験に固有である実験のパラメータを表 4.5 に示す。

記号で “protein9” と表示されるタンパク質の分布のみを正解の分布と比較した。

## 結果

17 回の試行での、世代毎の最良個体の適応度の推移を図 4.8 に示す。17 回の試行で見つかった最良個体を付録 B.1 に示す。付録 B.1 で示した最良個体の「合成-拡散-削除」ステップを重ねる様子を付録 B.2 に示す。図 4.9 に、最良個体の生成した、最も正解の分布に近いステップの時の分布を示す。

## 考察

まず、図 4.8 について考察する。この図は 17 回の試行での、世代毎の最良個体の適応度の推移を示している。図 4.8 を見ると、進化の進み具合という観点で 17 回の試行をいくつかのグループに分けることができる。適応度 2600 付近で進化が止ってしまっている試行群と、適応度 2200 付近で止っている試行群と、その間を移行中の試行、そして最後に適応度 1900 付近にまで飛び抜けている 1 つだけの試行が見られる。適応度 2600 付近に集まっている試行での最良個体は縞を 1 本生成し、適応度 2200 付近に集まっている試行での最良個体は縞を 2 本生成する。もし縞の本数を増やす進化が連続的に行われるならば、それぞれの試行の最良個体の適応度の推移の線は開始点 (適応度 3000、世代 0) から放射状になる。しかし各試行の推移の仕方はいくつかのグループに分けることができ、しかも階段状に推移している。このことから、縞の本数を増やす進化が断続的に行われたことが推測される。

つぎに、最良個体が「合成-拡散-削除」ステップを重ねる様子について考察する。正解の分布に最も近い分布は第 42 「合成-拡散-削除」ステップ目で得られた。この正解の分布に最も近い分布を形成した後、タンパク質の合成が過剰になって濃度がはね上がるという経過をたどる。このことは付録 B.2 の図 B.12 に見ることができる。これは *eve* を生成するルールが多いことによる。最良個体の 61 のルールうち、*eve* 自身の濃度を条件にして *eve* を合成するルールが 16 もある。その条件は *eve* の濃度がある一定の値以上なら *eve* を合成するというものである。つまりポジティブフィードバックである。このことが第 42 「合成-拡散-削除」ステップ目という早い段階で *eve* 縞を形成する一方、その後急速に正解の分布から離れていってしまうという結果をもたらしている。

一連の「縞形成の仕組みの進化シミュレーション」の実験では、第 4.3.2 節でも説明したように、「合成-拡散-削除」ステップのうち削除の段階で、各細胞で各タンパク質の量が 1 単位だけ削除されるようになっている。すなわち、現在の条件であるタンパク質を合成するルールが一つもなければ、そのタンパク質はステップを重ねるうちになくなってしまうことになる。現在の条件であるタンパク質を合成するルールが一つあるなら、拡散という処理を除けばそのタンパク質は一定の量を保つことができる。現在の条件であるタンパク質を合成するルールが二つあるなら、そのタンパク質の量が増えて行くことができる。したがって、*eve* 縞を形成するには最適化の過程で、異なる条件を持つ *eve*



を合成するルールが複数生成されなくてはならない。

また、得られた最良個体による縞形成の仕組みは、実際のショウジョウバエの *eve* 縞形成の仕組み (図 4.10 参照) とは全く異なる様式によるものであった。このことは、進化の過程で可能性としては有り得た縞形成の仕組みを探索し発見したと考えられる。

これらの考察から、次のような条件のもとで実験を試みる価値があると思われる。

- 早い「合成-拡散-削除」ステップで *eve* 縞が形成されないように、遅い「合成-拡散-削除」ステップからしか正解の分布との比較をしない。つまり「合成-拡散-削除」ステップをある程度重ねなければ個体を評価しない。
- 同じルールが多数存在することは、特定のタンパク質が短期間で多く合成され、急激に正解の分布から離れてしまう恐れがある一方、縞を形成するには必要である。ことことから、新しい遺伝的オペレータである「ルールの複製」を加えて最適化を試みる。「ルールの複製」オペレータは名前の通り、ランダムにルールを1つ選択し、それを複製して個体のルールの集合に加えるというものである。
- 実験によって、実際のショウジョウバエの *eve* 縞形成の仕組みとは異なった、進化の過程で可能性としては有り得た縞形成の仕組みを探索し発見したと考えられる。しかしより実際のショウジョウバエの *eve* 縞形成の仕組みに近いものを最適化で得られないか試みる。たとえば正解の分布に、*eve* 縞に限らず *eve* 縞の形成を助けるギャップタンパク質の分布を加えて比較し評価することなど。

### 4.4.3 実験 3

#### 目的

文献[8]によれば、*eve* 遺伝子はギャップ遺伝子の産物のタンパク質の濃度の低い場所で活性化される。したがって、*eve* の転写の起る場所は、ギャップタンパク質が高い濃度にある場所に挟まれる形によって決定される (図 4.10)。

この実験では、より実際のショウジョウバエの *eve* 縞形成の仕組みに近いものを最適化で得ることを試みる。実験2ではギャップタンパク質の分布については考慮せずにルールの集合の最適化を行ったが、この実験では *eve* 遺伝子の縞模様 (4 本) および、ギャッ

表 4.5: 実験2のパラメータ

| パラメータ                  | 値                       | 遺伝的オペレータ | 確率                  |
|------------------------|-------------------------|----------|---------------------|
| 個体数                    | 1000                    | 交差       | $12/30 = 0.4$       |
| 初期化時の木の深さ              | 3                       | 突然変異     | $2/30 \simeq 0.067$ |
| 初期化時のルールの数             | 20                      | 個体の複製    | $1/30 \simeq 0.033$ |
| トーナメントサイズ              | 7                       | ルールの交差   | $8/30 \simeq 0.267$ |
| 最大の「合成-拡散-削除」<br>ステップ数 | 500                     | ルールの追加   | $5/30 \simeq 0.167$ |
| 分布の比較をはじめるス<br>テップ数    | 0                       | ルールの削除   | $2/30 \simeq 0.067$ |
| タンパク質を表す記号で<br>使用されるもの | protein0 から<br>protein9 |          |                     |

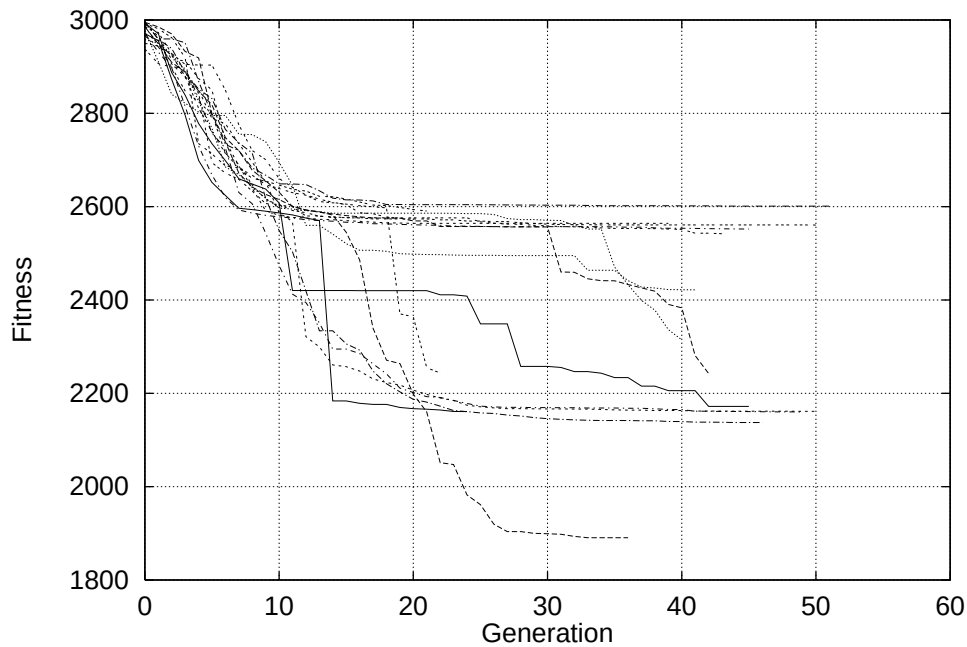


図 4.8: この図では実験2における合計17回の試行での、世代毎の最良個体の適応度の推移を示している。

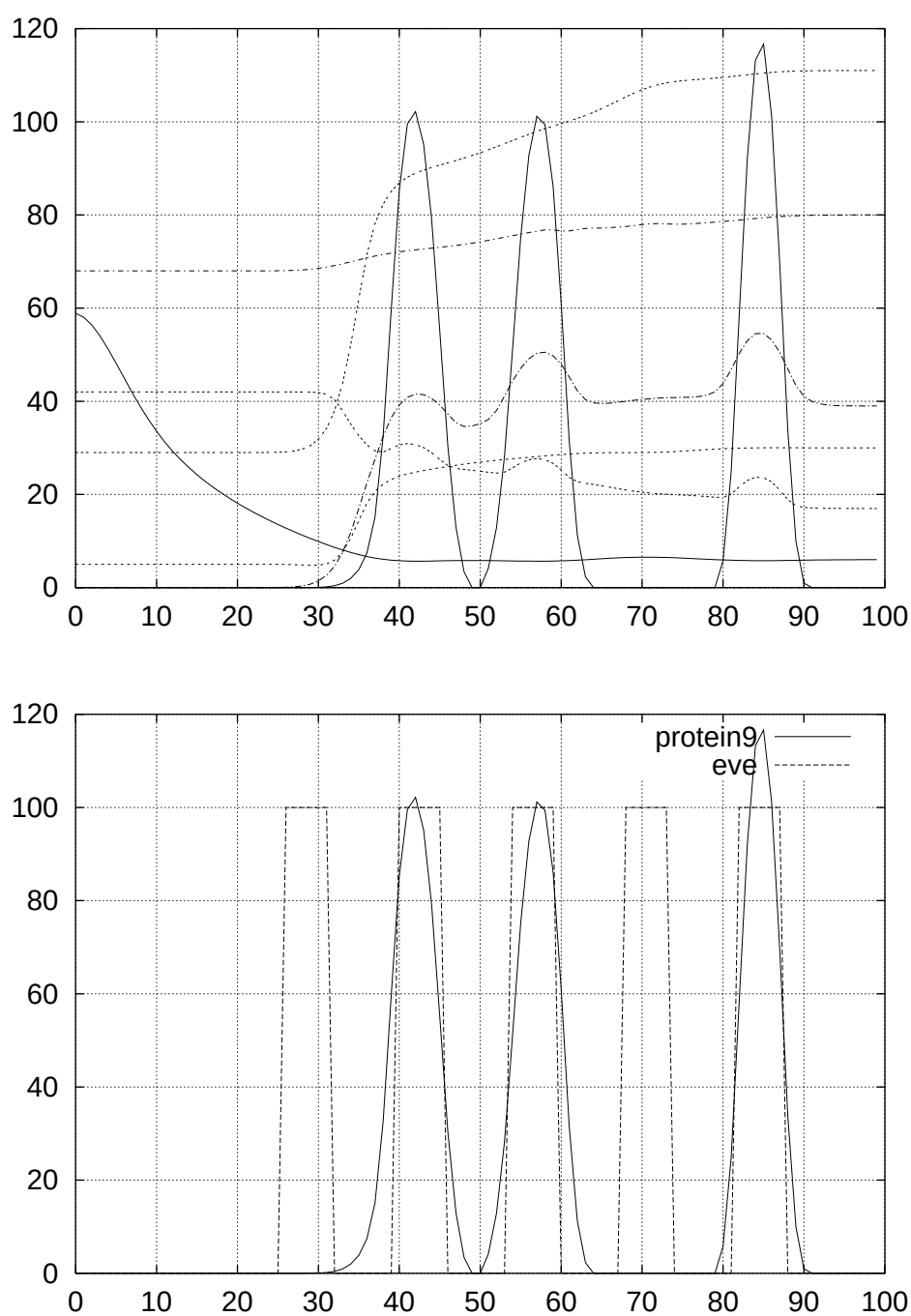


図 4.9: 実験 2 の最良個体の生成した、最も正解の分布に近いステップの時の分布。この分布は第 42 「合成-拡散-削除」ステップ目で得られた。上の図は、最良個体の評価に伴って生成されたすべての遺伝子調節タンパク質の分布を示している。下の図は、*eve* に相当する *protein9* を、正解の分布 (図中では “*eve*”) と比較しやすいように上の図から抜き出してきたもの。この図からは、GP による最適化によって 3 本綫を生成するルールが得られたことがわかる。

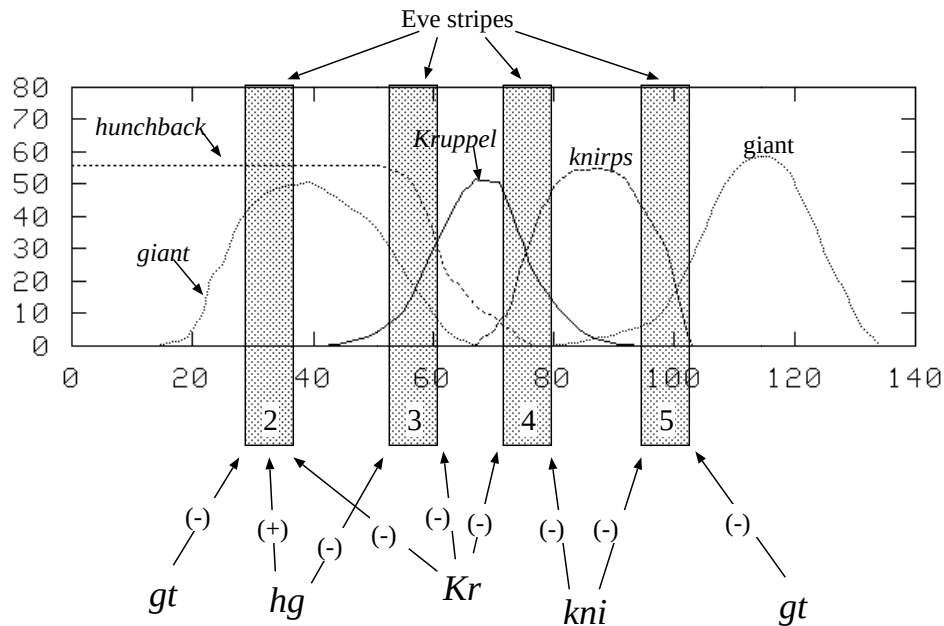


図 4.10: *even-skipped* 遺伝子の縞形成についての仮説。文献 [8] の Fig 14.23 より。図からはギャップタンパク質のそれぞれが *eve* にたいするリプレッサーとして働き、*eve* 縞はギャップタンパク質の分布の隙間に発現することが見て取れる。とくに縞 3, 4, 5 に顕著にそのことが現れている。

プタンパク質の分布を形成する規則の集合 (図 4.11) を得ることを試みた。実験 1 や 2 の時と違って、正解の分布の比較をするときに、複数のタンパク質について分布の比較を行いその差の合計を個体の適応度とした。

## 方法

この実験のパラメータを表 4.6 に示す。パラメータについて実験 1 や 2 と違う点は、最大のタンパク質の濃度が設定されていることと、分布の比較をはじめるステップ数が設定されていること、「規則の複製」オペレータ (第 4.3.5 節) を使用していることである。

ランダムに生成される規則によっては、ポジティブフィードバックがかかる。「合成-拡散-削除」ステップが進むことで、タンパク質の濃度が 100 以上になることがある。しかし、第 4.3.4 節「使用している記号」でも触れた通り、規則の条件式で使われる

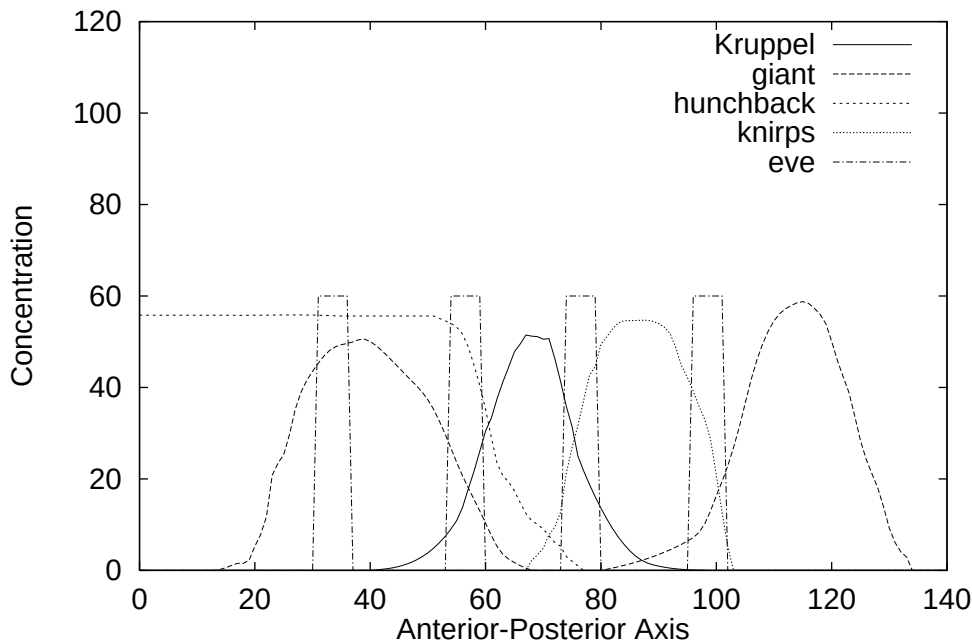


図 4.11: 実験 3 における目指す遺伝子調節タンパク質の分布

単精度浮動小数点数の範囲は 0 から 100 の間でランダムに設定される。つまりタンパク質の濃度が 100 以上になっても、減るのに時間がかかるだけで無駄になってしまう。というのは、第 4.3.2 節で触れた通り、「合成-拡散-削除」ステップの最後の段階で、各細胞で各タンパク質の量が 1 単位だけ削除される。そのため、あるタンパク質の濃度が 200 のとき、濃度が 100 まで戻るには 100 ステップ必要になる。この無駄になっていると思われるステップを避けるため、タンパク質が飽和する状況を擬似的に作り出すことを試みた。ルールを評価し IF 文の第 1 引数でしていされている条件が満たされていれば、IF 文の第 2 引数で指定されているタンパク質が生成されるようになっている。その際、生成される場所のタンパク質の濃度がパラメータ「最大のタンパク質の濃度」以上であれば、生成したタンパク質は捨てられるようになっている。

また、正解の分布と現在の分布の比較をしてルールの集合の適応度を決めるが、現在のステップ数がパラメータ「分布の比較をはじめるステップ数」に満たない場合、比較はされないようになっている。「分布の比較をはじめるステップ数」を設定したのは、早い「合成-拡散-削除」ステップの段階で適応度のピークを迎えてしまうような局所解に捕らわれることを避けるためである。

表 4.6: 実験 3 のパラメータ

| パラメータ                  | 値                       | 遺伝的オペレータ | 確率                   |
|------------------------|-------------------------|----------|----------------------|
| 個体数                    | 1000                    | 交差       | $16/61 \simeq 0.262$ |
| 初期化時の木の深さ              | 3                       | 突然変異     | $2/61 \simeq 0.033$  |
| 初期化時のルールの数             | 20                      | 個体の複製    | $1/61 \simeq 0.016$  |
| トーナメントサイズ              | 7                       | ルールの交差   | $30/61 \simeq 0.492$ |
| 最大の「合成-拡散-削除」<br>ステップ数 | 500                     | ルールの追加   | $6/61 \simeq 0.098$  |
| 分布の比較をはじめるス<br>テップ数    | 400                     | ルールの削除   | $4/61 \simeq 0.066$  |
| 最大のタンパク質の濃度            | 100                     | ルールの複製   | $2/61 \simeq 0.033$  |
| タンパク質を表す記号で<br>使用されるもの | protein0 から<br>protein9 |          |                      |

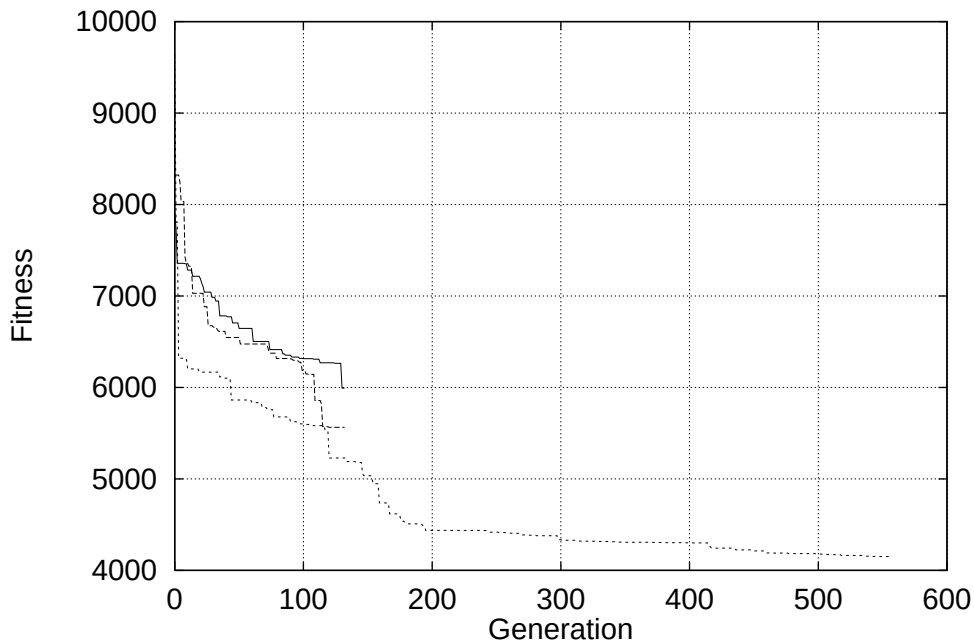


図 4.12: 実験 3 における世代毎の最良個体の適応度の推移

## 結果

3 回の試行での、世代毎の最良個体の適応度の推移を図 4.12 に示す。3 回の試行で見つかった最良個体を付録 C.1 に示す。付録 C.1 で示した最良個体の「合成-拡散-削除」ステップを重ねる様子を付録 C.2 に示す。

前に述べた通り、この実験では複数の遺伝子調節タンパク質との比較を行っている。一つの図 (図 4.13、図 4.14) では比較しにくいので、各遺伝子調節タンパク質ごとの図 (図 4.15、図 4.16、図 4.17、図 4.18、図 4.19) を以下に示す。

## 考察

*hunchback* の分布は、実験 3 の他の試行においても正解に近い分布が出現する。また、図 4.18 を見るとわかる通り、*knirps* や *giant* の分布についても正解の分布と良くにている。これらのような単純な分布を形成するルールは、比較的容易に生成することができると思われる。しかしながら肝心の *eve* はまったく合成されなかった。

*eve* 遺伝子の縞模様 (4 本) および、ギャップタンパク質の分布を形成するルールの集

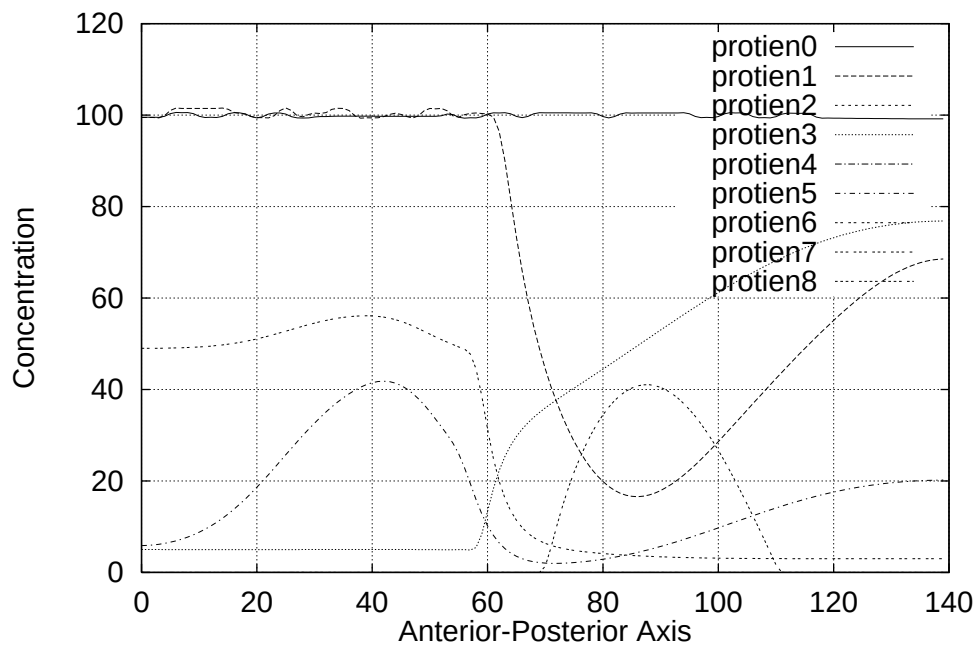
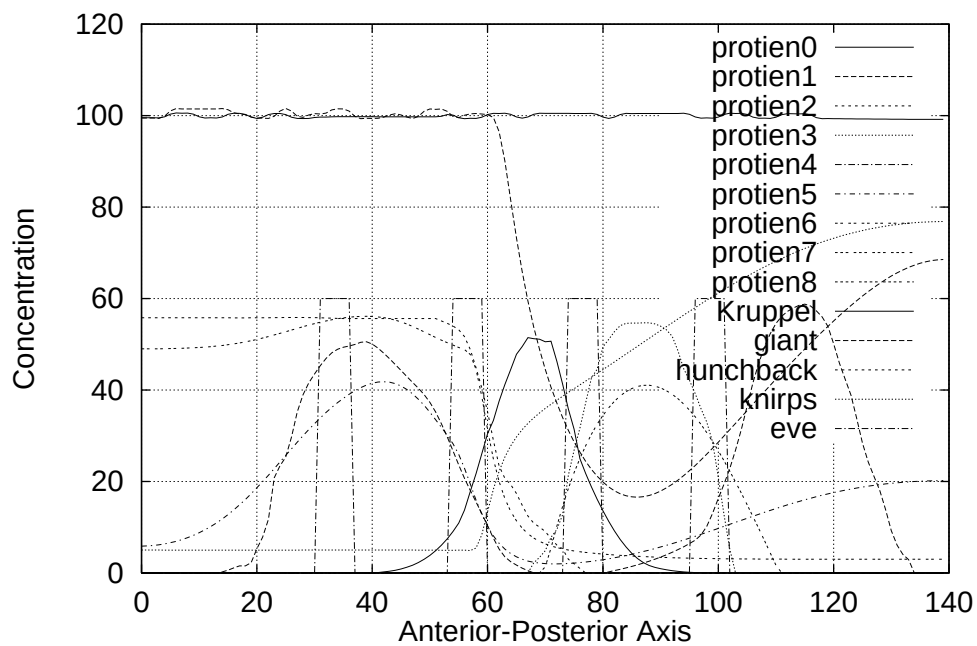


図 4.13: 実験 3 における最良個体の最良ステップ





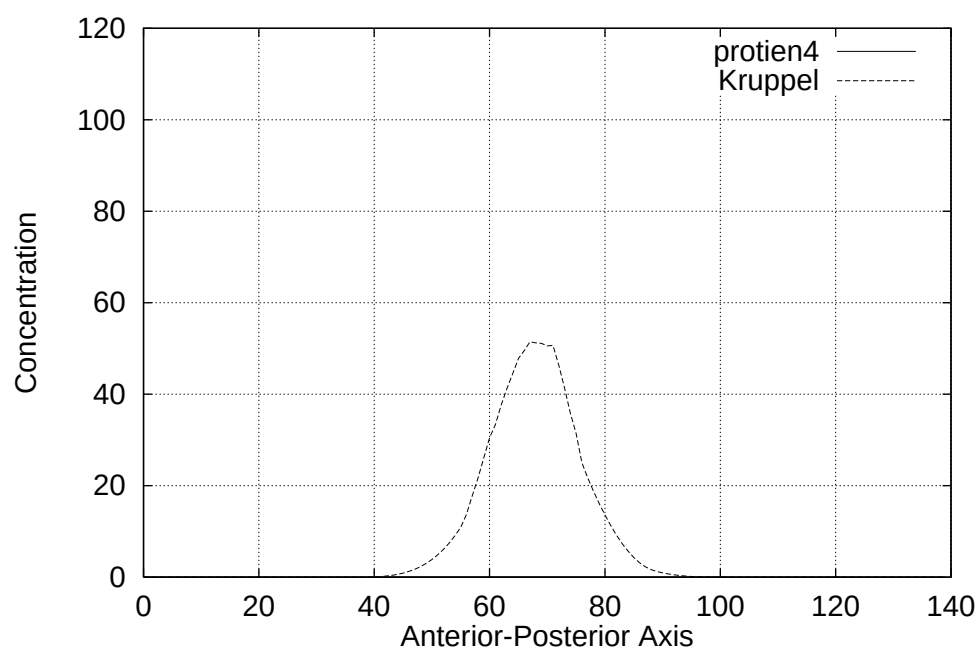
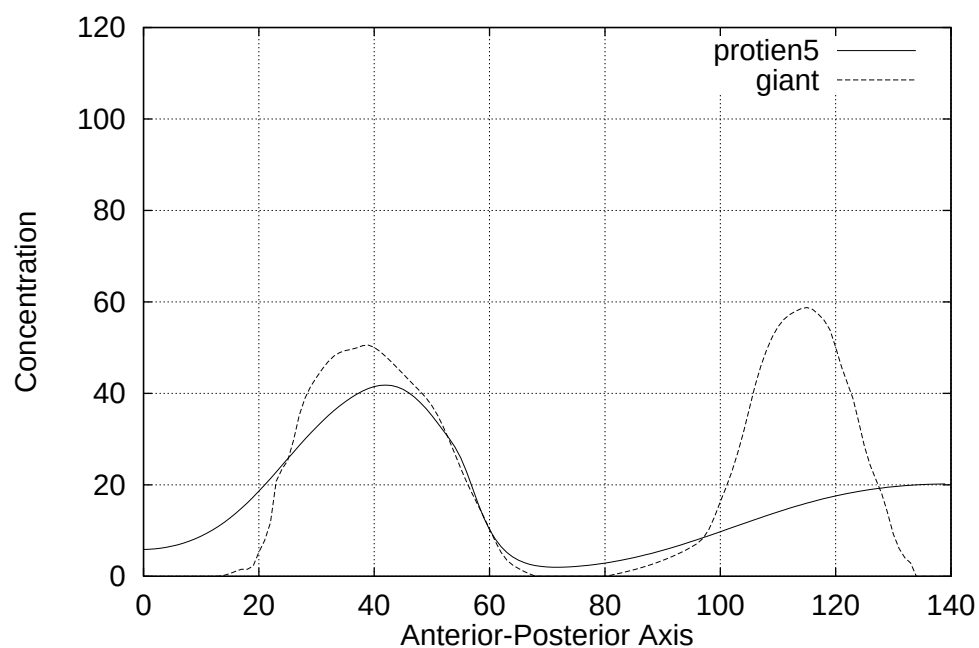


図 4.15: 実験 3 における最良個体の最良ステップ。 *Kruppel* は全く合成されていない。



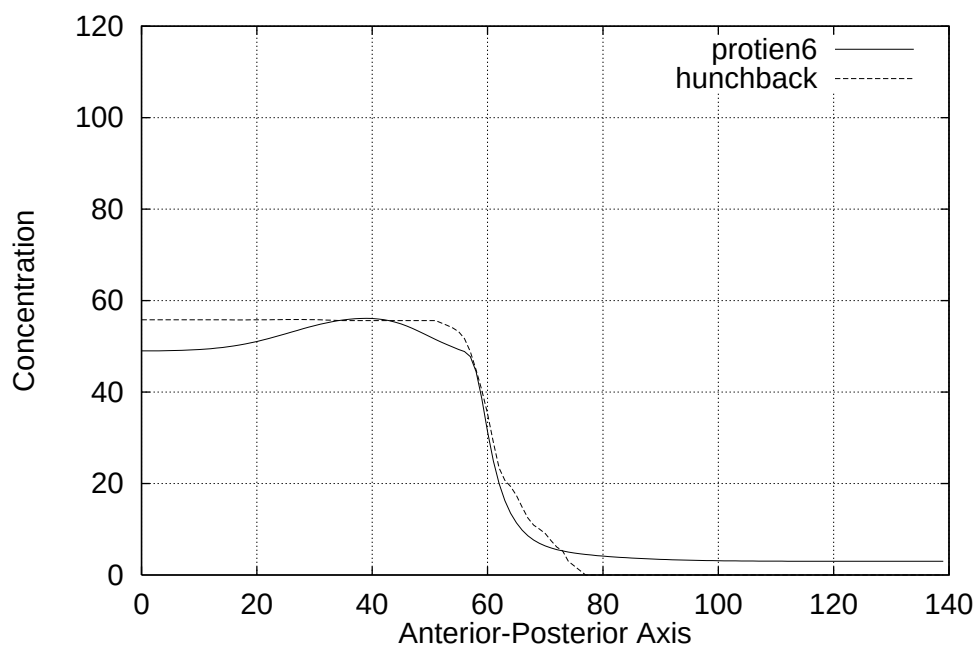
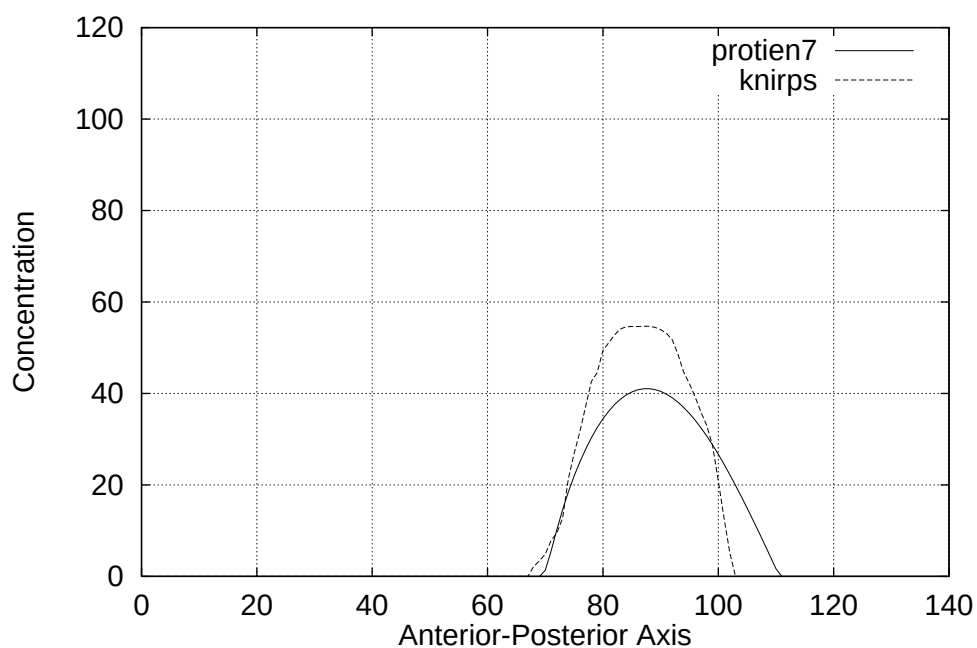


図 4.17: 実験 3 における最良個体の最良ステップ。ほぼ分布が一致していることが見て取れる。



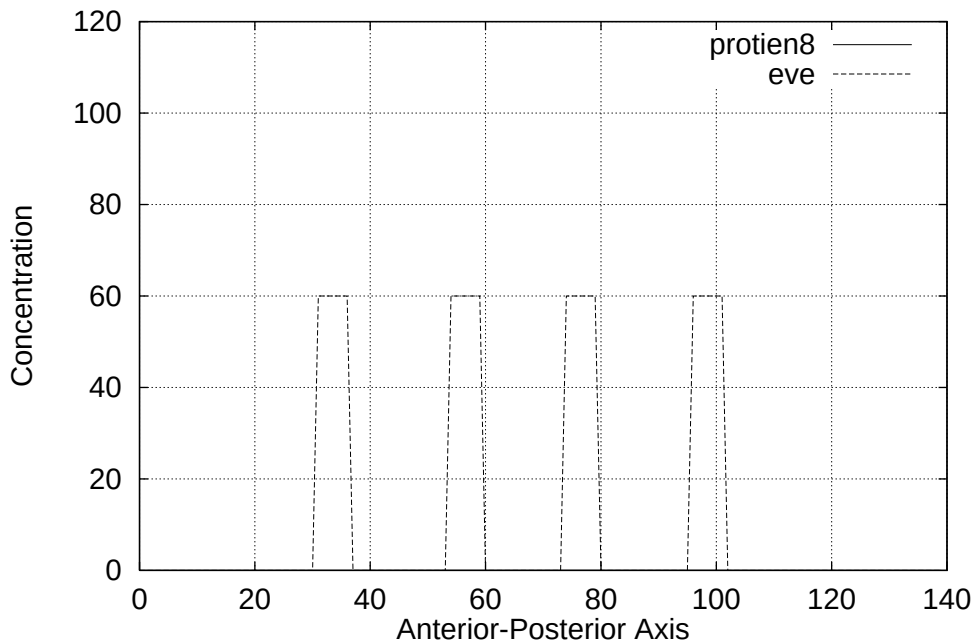


図 4.19: 実験 3 における最良個体の最良ステップ。eve は全く合成されていない。

合 (図 4.11) を得ようとした。実験 1 や 2 の時と違って、正解の分布の比較をするときに、複数のタンパク質について分布の比較を行いその差の合計を個体の適応度とした。けれども、ギャップタンパク質の単純な形の分布にマッチさせようと最適化が進み、局所解に入り込んでしまったものと思われる。または、実験 2 のときのように進化が断続的に進んでいるのかも知れない。まず最初にギャップタンパク質のような単純な形を形成するルールについての最適化が進み、最後に eve 縞の形成をするルールの最適化が行われるのかも知れない。しかし図 4.12 を見てもわかる通り、およそ 500 世代を経過しても 200 世代からほとんど適応度の伸びは観察されなかった。1 回の試行のみの結果で結論づけることはできないが、eve とギャップタンパク質の複数の分布との比較を行い評価する方法では、ギャップタンパク質の単純な形の分布にマッチさせようと最適化が進んでしまい、eve 縞の形成に向かっての最適化が行われにくいと推測することができる。

#### 4.4.4 実験 1, 2, 3 についてのまとめ

実験 1 では正解の分布 (図 4.5) によく一致させることができるルールの集合が得られた。実験 2 では実際のショウジョウバエの縞形成の仕組みとは全く異なるものの、3 本縞を形成するルールが得られた。実験 3 では、*hunchback* や *knirps* や *giant* などのあまり複雑な形でない分布 (図 4.17、図 4.18、図 4.16) にはよく一致させることができるルールの集合が得られた。

3 本以上の *eve* の縞のような複雑な形を持つ分布を形成するルールの集合を得られなかったことについて、シミュレーションに対する次のような問題点が考えられる。

**ただ単純に、パラメータの設定や使用する記号の集合が最適でないか。**

- 集団のサイズが足りない、または遺伝的オペレータの適用確率が最適でないなど。
- 得られたルールの集合 (付録 A.1 節、付録 B.1 節、付録 C.1 節など) のルールの特徴は、IF の条件部に AND や OR などの組み合わせが見られないことである。
  - 例外的に、付録 B.1 節の実験 2 の最良個体には膨大な条件部を持つルールがある。これらのことは実際の生物学的なデータとは異なる。
  - 実験で用いるパラメータの設定の仕方に問題があると思われる。たとえば次のようなパラメータ、「初期化時に生成する木の深さ」や、「突然変異時にランダムに生成する木の深さ」などに問題があると思われる。
  - 新たに次のような記号を導入する実験をやる価値があると思われる。その記号は強制的に条件部に AND を用いさせるものである。“(*IsConcentrationBetween*  $p$   $c1$   $c2$ )”。この条件式はもしタンパク質  $p$  の濃度が  $c1$  から  $c2$  の間であれば真を返し、そうでないときは偽を返す。

***bicoid* と *nanos* の分布が可変であること。**

- 母系効果遺伝子 *bicoid* の産物のタンパク質や *nanos* の産物のタンパク質は、*eve* 縞が形成されるときにも勾配を安定に保っていることが知られている。一連の実験では、*bicoid* タンパク質と *nanos* タンパク質を生成するようなルールを許してい

た。言い換えると、bicoid タンパク質の緩やかな勾配を描く分布を利用する個体は、bicoid タンパク質を合成し分布を保つようなルールを持っている必要がある。このことが不必要に探索空間を広げている可能性がある。

- この点をふまえて、次のような条件で実験を試みる価値があると思われる。
  1. bicoid と nanos は「合成-拡散-削除」ステップで削除されない。
  2. bicoid と nanos を示す終端記号を使わない (つまり bicoid と nanos を合成するルールが作られない)。
  3. ただし bicoid と nanos の濃度を参照する条件は作る。

#### 計算時間が足りないのではないか。

- 各実験での計算時間は短いものでは1週間ほど、長いものでは3週間ほどである。計算機資源が限られていることや落雷による停電の問題などで、1ヶ月以上の計算時間を得ることは事実上難しい。
- しかしながら、世代毎の最良個体の適応度の推移図 (図 4.8、図 4.12) を見てもわかる通り、やみくもに計算しても求める個体が見られるとは限らないと推測できる。

#### 4.4.5 実験 4

##### 目的

本実験は前節 (4.4.4 節) で取り上げた新たな条件を採り入れて、以前の実験とは違った結果が出るかどうかを観察するという目的で行われた。前節で取り上げた新たな条件は、簡潔にまとめると次のようなものであった。1. IsConcentrationBetween という記号を使用する。2. bicoid 遺伝子と nanos 遺伝子の産物の分布を安定させる。

## 方法

実験3ではギャップタンパク質の単純な形の分布へ合わせようと最適化が進んでしまい、肝心の *eve* 縞を形成するルールを得ることができなかった。そのためこの実験では再び *eve* 縞とのみ比較し適応度を算出する方法 (第 4.3.2 節参照) を採用した。

目的の節でも触れた通り、胚発生の基礎となる母系効果遺伝子 *bicoid* と *nanos* の産物の分布を安定化することで、*eve* 縞形成を促進する。このことは探索空間を狭めることになるかと推測した。

探索空間を狭めるためにルールの条件部を次のように固定した。“(IsConcentrationBetween  $p$   $c1$   $c2$ )”。この条件式はもしタンパク質  $p$  の濃度が  $c1$  から  $c2$  の間であれば真を返し、そうでないときは偽を返す。

実験で用いられるパラメータの値を微妙に変えた。たとえば実験3では初期化時のルールの数は20であったが、本実験では15に減らした。このことは計算時間の短縮を狙ったものである。また、タンパク質を表す記号で使用するものの数を9から8へ一つ減らした。さらに、遺伝的オペレータの適用確率を「交差」や「ルール交差」に比重をおくように変更した。これは遺伝的アルゴリズムや遺伝的プログラミングでは、一般的に突然変移のような遺伝的オペレータより交差のような遺伝的オペレータの適用確率が高い方が最適化が進みやすいことが知られているためである。

また、この実験では実験1, 2, 3とは世代の数えかたを変えた。実験1, 2, 3ではSteady State GPの数えかたを用いている。それは遺伝的オペレータを個体数分だけ適用したとき1世代とする数えかたである。この実験では便宜上、100回遺伝的オペレータを適用した時、1世代とした。そのため実験1, 2, 3とこの実験の世代数に関しての比較を単純には行う事ができない。

## 結果

2回の試行での、世代毎の最良個体の適応度の推移を図4.20に示す。2回の試行で見つかった最良個体を付録D.1に示す。付録D.1で示した最良個体の「合成-拡散-削除」ステップを重ねる様子を付録D.2に示す。最良ステップの分布と正解の分布を重ねた図を、図4.21に示す。

表 4.7: 実験 4 のパラメータ

| パラメータ                  | 値                       | 遺伝的オペレータ | 確率     |
|------------------------|-------------------------|----------|--------|
| 個体数                    | 2000                    | 交差       | 30/100 |
| 初期化時の木の深さ              | 3                       | 突然変異     | 5/100  |
| 初期化時のルールの数             | 15                      | 個体の複製    | 1/100  |
| トーナメントサイズ              | 7                       | ルールの交差   | 55/100 |
| 最大の「合成-拡散-削除」<br>ステップ数 | 500                     | ルールの追加   | 5/100  |
| 分布の比較をはじめるス<br>テップ数    | 400                     | ルールの削除   | 2/100  |
| 最大のタンパク質の濃度            | 100                     | ルールの複製   | 2/100  |
| タンパク質を表す記号で<br>使用されるもの | protein0 から<br>protein8 |          |        |

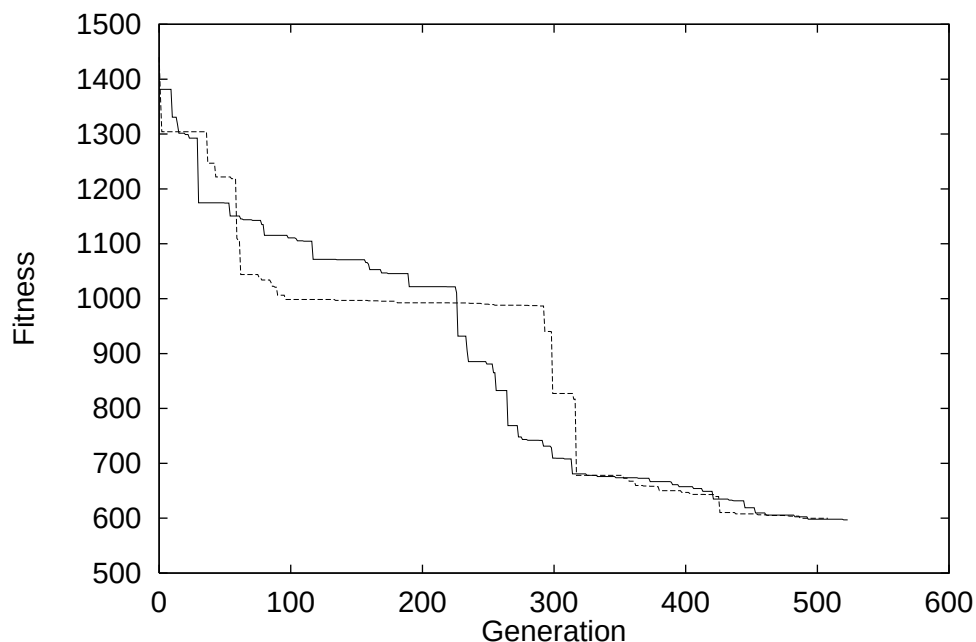


図 4.20: 実験 4 における世代毎の最良個体の適応度の推移。2 回の試行のうちどちらも 200 世代前半から 300 世代目にかけて急激に適応度を良くしているのが見てとれる。

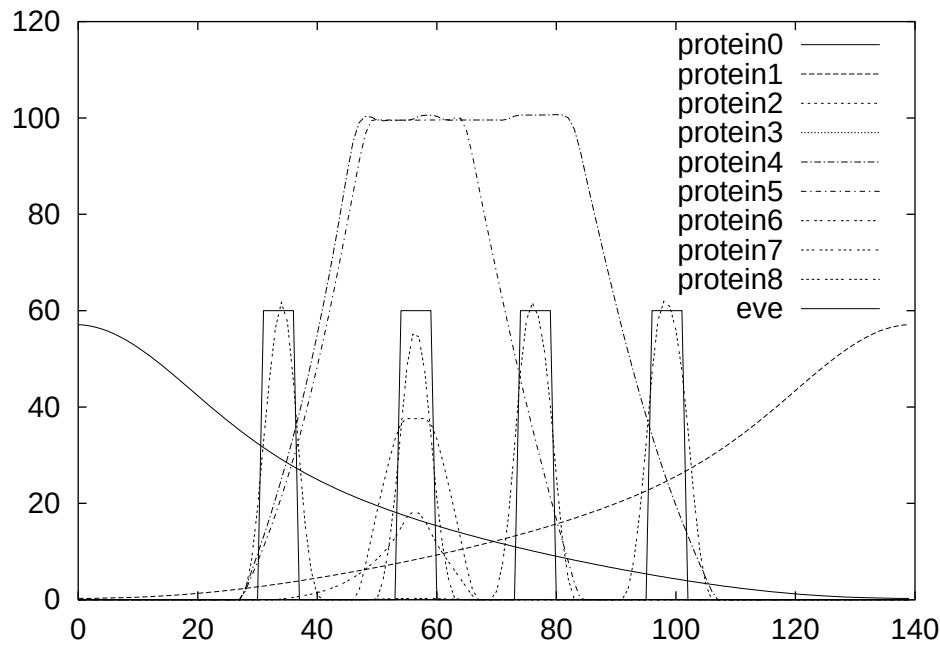


図 4.21: 実験 4 における最良個体の最良ステップ: 正解の分布と重ねたもの。図中の “eve” が正解の分布で、“protein8” が *eve* 遺伝子の産物に相当するものである。

## 考察

図 4.21 を見ると分かる通り、4 本縞をほぼ形成する個体を得ることができた。

正解の分布との比較が行われるステップを、400 から 500 「合成-拡散-削除」ステップ目に限った。このことで、得られた最良個体は実験 2 の時のように早い段階で正解に近い分布を形成して急速に正解から離れるということが無い。付録 D.2 を見ると分かる通り、およそ 240 「合成-拡散-削除」ステップ目から 4 本縞を形成し、そのまま 500 ステップ目までそれを維持している。

方法の節でも触れた通り、この実験の最大の特徴は母系効果遺伝子 *bicoid* と *nanos* の産物の分布を安定化することと、ルールの条件部にあるタンパク質の濃度が二値の間かどうかを判定するだけに固定した事である。この二つの特徴のうち、どちらが直接的に 4 本縞を形成するルールの集合を導き出しやすくしたかを調べるために追加実験を行った。それぞれの特徴を用いなくて最適化を行う実験をした。追加実験 1 は母系効果遺伝子の産物の分布を安定化するが、ルールの条件部は実験 2 や実験 3 の時と同じ記号の集合を使用した (つまり条件部を、あるタンパク質の濃度が 2 値の間にあるかどうかで固



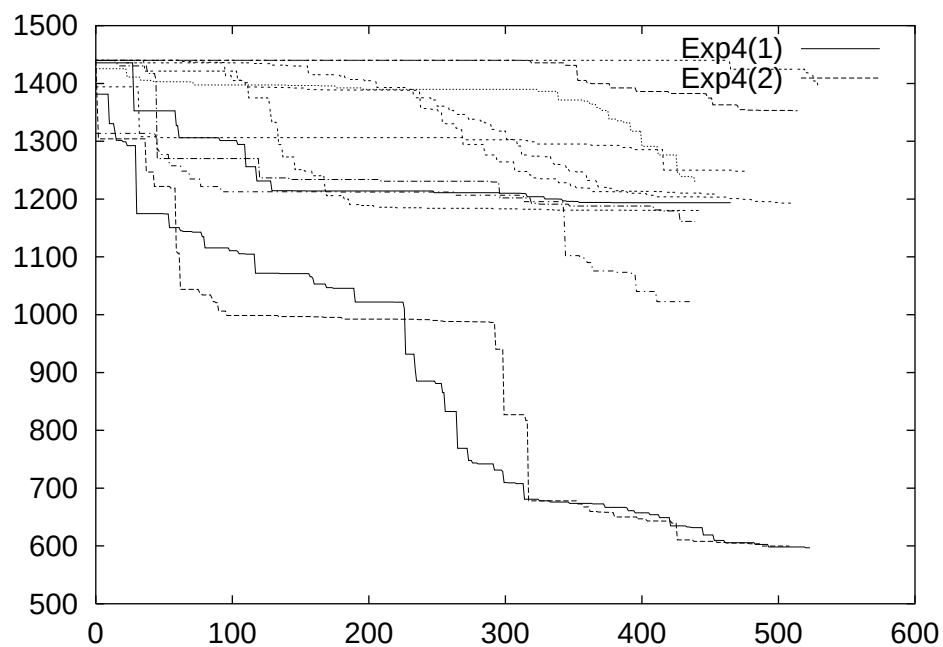


図 4.22: 世代毎の最良個体の適応度の推移 (実験 4 とその追加実験)。Exp4(1) と Exp4(2) が実験 4 の世代毎の最良個体の適応度の推移である。その他の線は追加実験の世代毎の最良個体の適応度の推移である。実験 4 と違って、追加実験では 4 本綫を形成する個体を得る事はできなかった。

表 4.8: 実験 4 で最良個体を更新した遺伝的オペレータと、それぞれの回数および割合。実験中、今まで最良個体とされていたものより、優れた個体を生成した遺伝的オペレータをログから抜き出した。回数の合計は 123 回である。割合は回数を 123 で割ったものである。

| 遺伝的オペレータ名 | 回数 | 割合     | 遺伝的オペレータ名 | 回数 | 割合     |
|-----------|----|--------|-----------|----|--------|
| 交差        | 29 | 23.6 % | ルールの交差    | 73 | 59.3 % |
| 突然変異      | 4  | 3.3 %  | ルールの削除    | 1  | 0.8 %  |
| ルールの追加    | 6  | 4.9 %  | ルールの複製    | 10 | 8.1 %  |

定しない)。追加実験 2 は逆に、母系効果遺伝子の産物の分布を安定化せず、条件部をタンパク質の濃度が 2 値の間にあるかどうか固定した。追加実験 3 は、どちらの特徴も用いずに行った。図 4.22 に、実験 4 の世代毎の最良個体の適応度の推移と各追加実験の世代毎の最良個体の適応度の推移を示す。この図からは、追加実験では実験 4 で見られた急激な進化が追加実験ではできなかったことが見てとれる。追加実験では実験 4 で得られた 4 本縞を形成する個体を得る事ができなかった。これらの実験結果から、次のようなことを結論づける事ができる。母系効果遺伝子 *bicoid* と *nanos* の産物の分布を安定化することと、ルールの条件部のあるタンパク質の濃度が二値の間かどうかを判定するだけに固定したことが、4 本縞を形成する個体を最適化によって得る事を促進した。

この実験では詳細なログを記録したので、それについて分析を試みた。

どの遺伝的オペレータが実験中に最良個体を更新したかを調べると、表 4.8 のような分布になった。

遺伝的オペレータ「ルールの複製」に関しては適用確率 2 % の約 4 倍の 8.1 % も、世代の最良個体を生成するのに貢献している。これは、同じ適用確率 2 % の遺伝的オペレータ「ルールの削除」の割合が 0.8 % であることと対照的である。実験 2 の考察でも触れたとおり、同じルールが多数存在することは、特定のタンパク質が短い期間に多く合成され、急激に正解の分布から離れてしまう恐れがある一方、縞を形成するには必要である。というのは、「合成-拡散-削除」ステップのうち削除の段階で各細胞で各タンパク質の量が 1 単位だけ削除されるようになっていて、あるタンパク質を生成するルールが 2 つ以上無ければそのタンパク質は「合成-拡散-削除」ステップが進んでも濃度を増やす事ができないからである。実験 4 では、正解の分布との比較が行われるステップ

を、400 から 500 「合成-拡散-削除」ステップ目に限った。そのため、「ルールの複製」の悪い結果である、急激に正解の分布から離れてしまうような個体は最適化の初期の過程で排除され、上述のような「ルールの複製」の必要な面が強調され、世代毎の最良個体を輩出させたのだと推測できる。

また、「ルールの交差」は適用確率 55 % であったが最良個体を更新した割合は 59.3 % であった。これは適用確率 30 % の通常の「交差」オペレータが最良個体を更新した割合が 23.6 % であった事と対照的である。このことは、一連の *eve* 縞形成の実験での個体の形式がルールの集合と言う形式をとり、通常の遺伝的プログラミングの個体のような単一木構造をとっていないことから来ているものと思われる。単一の木構造の場合、遺伝的プログラミングの伝統的な遺伝的オペレータの「交差」は、部分木の入れ換えを行うので、遺伝型の混ぜ合わせと言う点でその効果を発揮する。しかしルールの集合の場合、部分木の入れ換えは単一のルールの部分木の入れ換えとしかならない。ルールの集合すなわち個体全体として見た遺伝型の混ぜ合わせ、と言う点では効果が薄くなる。「ルールの交差」では複数のルールが入れ替わり、遺伝型の混ぜ合わせがより進みやすい。このことから、一連の *eve* 縞形成の実験では、遺伝的プログラミングの伝統的な遺伝的オペレータの「交差」より、「ルールの交差」が有効であったと推測できる。

次に、実験 4 の 2 回の試行のうち 100 世代目から 300 世代目まで、最良個体の更新が起らなかった方の試行 (図 4.20 の点線で示されている方) での、急激な進化について考察する。

図 4.20 の点線で示されている最良個体の適応度の推移を見ると、特徴的なのは 300 世代のはじめで急激に適応度を良くした点である。この急激な進化は 300 世代目と 318 世代目で起きている。300 世代目では適応度が 939.7 から 827.1 まで良くなっていて、318 世代目では適応度が 816.8 から 678.0 まで良くなっている。

この実験では、詳細なログをとってあるので、この急激に適応度が良くなった上述の 2 個体の由来を調べることができた。それぞれの個体の親や、親の親と家系を 5 世代までさかのぼって調べてることができた。それぞれの個体やその親たちを生成した遺伝的オペレータの種類とその適用回数を、表 4.9 に示す。

表 4.9 を見て気づくことは、ここでもやはり遺伝的オペレータ「ルールの複製」の適用回数が多いことである。「ルールの複製」の適用確率 2 % に対し、割合は 10.1 % と

表 4.9: この表では、実験 4 での急激に適応度を良くした 2 個体の家系を 5 世代前まで調べ、どの遺伝的オペレータが何回適用されたかを示す。合計は 79 回で、回数を合計で割ったものを割合として示した。

| 遺伝的オペレータ名 | 回数 | 割合     | 遺伝的オペレータ名 | 回数 | 割合     |
|-----------|----|--------|-----------|----|--------|
| 交差        | 27 | 34.2 % | ルールの交差    | 34 | 43.0 % |
| 突然変異      | 3  | 3.8 %  | ルールの削除    | 0  | 0.0 %  |
| ルールの追加    | 7  | 8.9 %  | ルールの複製    | 8  | 10.1 % |

なっている。また、逆に遺伝的オペレータ「ルールの交差」の割合が 43.0 % となっていて、適用確率 55 % と比べて少なくなっている。さらに、遺伝的オペレータ「交差」は適用確率 30 % の割りに、割合は 34.2 % と多くなっている。

また、急激に適応度を伸ばした 2 個体のを生成したオペレータは、「ルール交差」と「交差」であった。

このことについて、次のような推測が可能である。図 4.20 を見るとわかる通り、点線で示されている方の試行では、100 世代目から 300 世代目まで、ほとんど目に見える進化が行われていない。おそらくこの間に、「突然変異」系の遺伝的オペレータである「突然変異」、「ルールの追加」、「ルールの複製」の成果が累積したのであろう。そして GA でいうところの、有効な「積み木」ができあがり、「ルール交差」や「交差」などによってより良い個体が出来たのであろう。

## 4.5 検証: 位置シグナル強度のわずかな差異に左右されないか

実験 4 で得られた最良個体が、位置シグナル強度のわずかな差異に左右されない段階的なパターン形成の仕組みを実現しているかどうかを調べる実験を行った。ここではその実験方法および結果と考察を述べる。

### 4.5.1 方法

実験4で得られた最良個体を評価する際に、母系効果遺伝子 *bicoid* および *nanos* の産物の初期分布をそれぞれ変化させた。変化は初期分布全体を定数倍する方法で行った。

### 4.5.2 結果

Bicoid タンパク質および Nanos タンパク質の初期分布を、ともに定数倍した場合、Bicoid タンパク質の初期分布のみ定数倍した場合、Nanos タンパク質の初期分布のみ定数倍した場合、について結果を表 4.10 に示す。

表 4.10 に示されている適応度はとても悪いものになっている。しかし、Bicoid と Nanos の両方の初期分布を +12% から -3% 定数倍した場合でも 4 本縞が確認された。適応度が悪くなったのは、正解の分布と比べて左右に縞がずれたためである。適応度の算出方法は、評価した結果の分布と正解の分布の差であるため、左右にずれた場合に 4 本縞が発現していても悪いものになってしまう。

### 4.5.3 考察および議論

実験4で得られた最良個体は、位置シグナル強度のわずかな差異に左右されない頑健なパターン形成の仕組みを実現していた。このことは、表 4.10 に示されている通り、Bicoid と Nanos の初期分布全体を +12% から -3% 定数倍しても 4 本縞を形成したことから明らかである。さらに、Nanos の初期分布の全体のみでは、+20%定数倍しても 4 本縞が得られたし、Bicoid の初期分布の全体のみでは、+30% 定数倍しても 4 本縞が得られた。

では、なぜ頑健なパターン形成の仕組みが得られたのか？ 実験では1つの初期分布のみ使用して評価した。すなわち、初期分布に誤差が生じるような環境で評価し、最適化したのではない。このような環境で最適化を行えば、初期分布のわずかな差異に左右されない個体を得られてもおかしくない。しかし、実験では1つの初期分布のみ使用して評価し最適化を行ったにもかかわらず、初期分布のわずかな差異には左右されない頑健なパターン形成の仕組みを得ることができた。

表 4.10: 初期分布を定数倍して評価した結果

(a) Bicoid と Nanos の初期分布をともに定数倍して評価した結果

| 定数倍の割合 | 適応度     | 観察された縞の本数 |
|--------|---------|-----------|
| 15 %   | 2396.58 | 3 本から 4 本 |
| 12 %   | 1799.60 | 4 本       |
| 10 %   | 1449.68 | 4 本       |
| 5 %    | 837.04  | 4 本       |
| 3 %    | 669.44  | 4 本       |
| 2 %    | 645.19  | 4 本       |
| 0 %    | 596.51  | 4 本       |
| -2 %   | 687.87  | 4 本       |
| -3 %   | 798.25  | 4 本       |
| -5 %   | 1281.97 | 3 本から 4 本 |
| -7 %   | 1692.43 | 3 本       |

(b) Bicoid の初期分布だけを定数倍して評価した結果

| 定数倍の割合 | 適応度     | 観察された縞の本数  |
|--------|---------|------------|
| 35 %   | 1406.99 | 3 本 から 4 本 |
| 30 %   | 1269.13 | 4 本        |
| 25 %   | 1049.35 | 4 本        |
| 20 %   | 1029.06 | 4 本        |
| 15 %   | 928.33  | 4 本        |
| 12 %   | 838.68  | 4 本        |
| 10 %   | 784.03  | 4 本        |
| 5 %    | 660.83  | 4 本        |
| 3 %    | 607.81  | 4 本        |
| 2 %    | 612.86  | 4 本        |
| 0 %    | 596.51  | 4 本        |
| -2 %   | 642.29  | 4 本        |
| -3 %   | 658.52  | 4 本        |
| -5 %   | 739.73  | 4 本        |
| -7 %   | 831.27  | 4 本        |
| -10 %  | 1318.43 | 3 本        |

(c) Nanos の初期分布だけを定数倍して評価した結果

| 定数倍の割合 | 適応度     | 観察された縞の本数  |
|--------|---------|------------|
| 25 %   | 2377.92 | 3 本 から 4 本 |
| 20 %   | 2022.43 | 4 本        |
| 15 %   | 1558.13 | 4 本        |
| 10 %   | 1184.08 | 4 本        |
| 5 %    | 803.87  | 4 本        |
| 3 %    | 693.95  | 4 本        |
| 2 %    | 653.27  | 4 本        |
| 0 %    | 596.51  | 4 本        |
| -2 %   | 679.59  | 4 本        |
| -3 %   | 726.57  | 4 本        |
| -5 %   | 839.69  | 4 本        |
| -7 %   | 1025.82 | 4 本        |
| -10 %  | 1586.85 | 3 本 から 4 本 |

先ほどの疑問、「なぜ頑健なパターン形成の仕組みが得られたのか？」について調べる手段として、次のような実験が考えられる。初期分布のわずかな差異によって縞形成が行われなくなる個体を生成する進化シミュレーションのパラメータおよび設定を探す。得られたパラメータおよび設定と、実験4のパラメータおよび設定を比較する。そして、頑健な仕組みを生成する原因となったパラメータまたは設定を特定する。しかしながら、本研究では時間的な制限からこのような実験を行うことができなかった。

## 4.6 結論

本研究では、次のような知見を得た。単純な記号の組合わせでタンパク質を合成するルールを構成し、遺伝的プログラミングにより最適化するだけで、縞形成の複雑なダイナミクス (付録を参照のこと) が観察できるルールの集合を得ることができることである。さらに、ある特定の条件のもとで最適化を行い、4本縞を形成するルールの集合を得ることができた。その条件とは 1. 初期分布を安定化させる、2. ルールの条件部は「あるタンパク質の濃度が上限値と下限値の間であるかどうか」に固定することである。

そのような条件で得られた最良個体は、初期分布がわずかに違った場合でも、正解として与えた4本縞に似た縞を発現した。つまり、位置シグナル強度のわずかな差異に左右されない頑健なパターン形成の仕組みを得ることができた。

また、進化シミュレーションの特徴を活かし、急激に適応度を伸ばした個体について家系を調べ、どの遺伝的オペレータが効果的に働いたかを探ることができた。

## 4.7 今後の課題

拡散の方法は、各細胞の各遺伝子調節タンパク質の  $1/4$  づつを両隣の細胞に渡し、最左と最右の細胞は各遺伝子調節タンパク質の  $1/2$  を隣の細胞に渡すという方法であった (第 4.3.2 節参照)。タンパク質ごとに異なる拡散定数をもつことが知られている。すなわち、タンパク質ごとに異なる拡散の仕方をするということである。しかし一連の実験ではすべてのタンパク質の拡散の仕方を同じにした。これらのこと (タンパク質の拡散の方法それ自体、または拡散の仕方がタンパク質のどれも同じであること) に問題があ

る可能性も捨てきれない。

実験4では特定の条件(1. 初期分布を安定化させる、2. ルールの条件部は「あるタンパク質の濃度が上限値と下限値の間であるかどうか」に固定する)のもとでは4本縞を得ることができる事を示した。実験2のような自由な条件部や、実験3のギャップタンパク質の分布とも比較するなどのもとでは最高でも3本縞しか得られなかった。しかしながら、図4.8で示した実験2の世代ごとの適応度の推移や、図4.12で示した実験3の世代ごとの適応度の推移を見るかぎり、最適化が行われていることは確かである。このことから、ルールの条件部を固定せず、記号の組合せが完全に自由な環境で、頑健なパターン形成の仕組みを創発させることが期待できる。

また、実験4では単一の初期分布のみ使用して評価し最適化を行ったにもかかわらず、複数のわずかづつ異なる初期分布でも4本縞を発現する(すなわち、初期分布のわずかな差異に左右されない)頑健なパターン形成の仕組みを得ることができた。この理由を探ることも今後の課題である。

さらに言えば、プログラムの実行時間の長さが問題となった。実験4では1回の実験で3週間以上かかった。実験に長い計算を要する事が、条件を変えて繰り返し実験することを妨げた。そのため、プログラムを最適化および並列化することが今後の課題となった。



## 第 5 章

# PC を利用した並列システム: mugyu

### 5.1 はじめに

「縞形成の仕組みの進化シミュレーション」を行うのには、多くの計算機パワーを使う。1つのシミュレーションが2週間以上かかることは珍しくない。多数の異なるパラメータでシミュレーションを行うために、多数の CPU が必要となる。

知識科学教育研究センターの電算室や研究室にあまり使用されていないパーソナルコンピュータ (PC) があった。これらの PC をシミュレーションに活用することを考えた。これらの PC には Windows がインストールされている。しかし、シミュレーション用のプログラムは、Linux や SunOS などの UNIX 系オペレーティングシステム (OS) でのみコンパイルおよび実行可能である。シミュレーションを行うために Linux を PC へインストールし、授業で Windows が使用される前に Linux を削除するには手間がかかる。そのため、PC のハードディスクにはまったく手をつけず、フロッピーで起動する Linux システムを作成した。この方式であれば、PC を再起動すれば、また Windows 機としてすぐさま稼働可能である。

この章では本研究で作成した、ハードディスクをいっさい使用せずフロッピーで起動する Linux システム (“mugyu” 「ムギュー」と読む) について説明する。まず 5.2 節で mugyu システムの目的を示す。次に 5.3 節で mugyu システムの構成についての概要を述べる。そして 5.4 節でシステムのインターフェースと使用例を紹介する。最後に 5.5 節

で mugyu システムの仕組みを説明する。

また、本章での説明中で UNIX コマンド名やシステムコール、UNIX のユーザ関数名、ライブラリ名を引用するとき “command(n)” と示した。command はコマンド名である。n には、そのコマンドの説明の属する man ページ (マニュアルページ) のセクションである。

## 5.2 目的

- 実験に必要な、UNIX 的な OS 上での多数の強力な計算パワーを得ること
- ただし、ハードディスクに何も書き込まない (すぐに Windows が稼働可能であること)
- システムに加わっている計算機の負荷状態を手軽に調べられること
- たくさんのジョブを手軽に投げられること
- 走っているジョブのリストを手軽に閲覧できること

## 5.3 システム構成の概要

上述の目的を達成するために、システムを3つの層からなる構成にした。第1の層は、ハードディスクを使用せず Linux オペレーティングシステムを稼働させることを実現する層である。すなわち、本システムのために作成した mugyu システムの起動用フロッピーディスクの内容物である。第2の層は、ジョブを投入するためのプログラムや、他のホストからのリクエストに答えてジョブを実行したりするプログラムからなる。第3の層は、第2の層に対するウェブインターフェースである。第3の層は、ウェブページからジョブを投入したり、ジョブを途中で中断したり、現在実行中のジョブを閲覧することを実現する。

また、mugyu システムをクライアント-サーバシステムという観点からみることもできる (図 5.1)。サーバの主な役割は、クライアントたちへ NFS (Network File System) 経由でディスクスペースを提供することである。また、サーバはクライアントに、起動

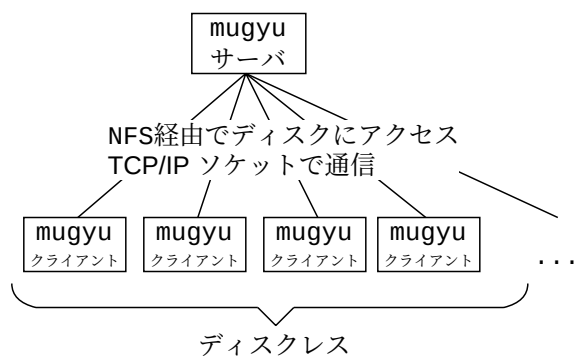


図 5.1: mugyu システムのクライアントとサーバの関係

用フロッピーディスクに入りきらなかったライブラリやプログラムを提供する。クライアントの役割はジョブを処理することである。

## 5.4 システムのインターフェース

### 5.4.1 システムの準備時

ここでは第1の層のインターフェース、すなわちシステムの立ち上げ時のインターフェースについて説明する。

クライアントの準備は、mugyu クライアント用のブートイメージが入った起動用フロッピーディスクを使って PC を起動し、設定画面で mugyu サーバの IP アドレスなどを設定するだけである。起動用フロッピーディスクは2枚組で、boot フロッピーディスクと root フロッピーディスクからなる。数多くの PC を mugyu クライアントにするため、クライアントの準備は簡略なものであることを目指した。また、あらかじめいくつかの項目にはデフォルト値を設定した。以下に、立ち上げ時に入力を求められる項目の一覧を挙げる。

- **mugyu サーバのホスト名:** (デフォルト値が設定されている)
- **mugyu サーバの IP アドレス:** (デフォルト値が設定されている)
- **クライアントのホスト名**

- クライアントのドメイン名
- クライアントの IP アドレス
- クライアントのネットマスク: (デフォルト値が設定されている)
- ゲートウェイアドレス: (デフォルト値が設定されている)
- ネームサーバの IP アドレス: (デフォルト値が設定されている)
- NTP サーバのホスト名: (デフォルト値が設定されている)
- NIS サーバのホスト名: (デフォルト値が設定されている)
- NIS ドメイン名: (デフォルト値が設定されている)

このように、デフォルト値が設定されているものが多く、多数の PC をクライアント化するのに便利になっている。

また、サーバの準備のインターフェースについては特に用意していない。サーバ側の用意としては、HTTP(Hyper Text Transfer Protocol) サーバを立ち上げるなどの準備が必要である。

## 5.4.2 システムの利用

### 各クライアントの負荷の状態を調べる

“lsload” コマンドが利用可能である。

```
ks27w0f17:~% lsload
```

| HOST      | USER | NICE  | SYS  | IDLE | 1m   | 5m   | 15m  |      |     |
|-----------|------|-------|------|------|------|------|------|------|-----|
| ks27w0f13 | 0.0% | 97.3% | 0.0% | 2.7% | 0.99 | 0.97 | 0.91 | 2/21 | 422 |
| kspc99h02 | 0.0% | 97.0% | 0.0% | 3.0% | 0.99 | 0.97 | 0.91 | 2/21 | 408 |
| kspc99h03 | 0.0% | 97.2% | 0.0% | 2.8% | 0.99 | 0.97 | 0.91 | 2/21 | 432 |

lsload コマンドの出力の各項目について説明する。“HOST” は、ホスト名を示す。“USER”、“NICE”、“SYS”、“IDLE” は各スペースにおける CPU の使用率を示す。

“1m”、“5m”、“15m” は、それぞれ1分、5分、15分あたりの実行中のジョブの数の平均である。

## ジョブの投入

“bsub” コマンドを用いてジョブの投入を行うことができる。

```
ks27w0f17:~% bsub - hgwr@jaist.ac.jp . ../test/multiple 100
submitted to host 'ks27w0f17'.
Command: ../test/multiple 100
JobID: 1
cwd: /mnt/space/hgwr/mugyu/src/mugyu_system/tmp
stdoutFile: ./hgwr_ks27w0f17_0_2000_0115_054112.stdout
stderrFile: ./hgwr_ks27w0f17_0_2000_0115_054112.stderr
```

“bsub” コマンドはコマンドライン引数に、mugyu システムのアカウント名、ジョブが終了したことを通知する先のメールアドレス、結果を出力するディレクトリ、ジョブであるプログラム、そのプログラムへのコマンドライン引数をとる。mugyu システムのアカウント名を持っていない場合には、第1引数は“-” でよい。メールによる通知が不要な場合は、第2引数は“-” でよい。mugyu システムのデフォルトの出力ディレクトリが用意されており、そちらを使用したい場合には第3引数は“-” と指定する。

ジョブが正常に投入されると、上のような出力がなされる。ジョブが投入されたホスト、ジョブとして投入されたコマンド、Job の ID 番号、作業が行われるディレクトリ、どのファイルにジョブの標準出力と標準エラー出力が保存されるかが表示される。

## 実行中のジョブの閲覧

“lsjobs” コマンドを用いて、現在実行中のジョブの一覧を見ることができる。

```
ks27w0f17% lsjobs
Host: ks27w0f17
user, mugyu_account, mail_address, jobID, pid, submit_time, from_host,
```

```
stdoutFileName, stderrFileName, command
hgwr, -, -, 5, 10073, Sat Jan 15 14:55:53 2000, 150.65.80.37, /usr/local/var/mugyu/output/hgwr_ks27w0f17_4_2000_0115_055553.stdout, /usr/local/var/mugyu/output/hgwr_ks27w0f17_4_2000_0115_055553.stderr, /mnt/space/hgwr/mugyu/src/mugyu_system/test/multiple, 100
```

Host: kspc99h03

```
user, mugyu_account, mail_address, jobID, pid, submit_time, from_host,
stdoutFileName, stderrFileName, command
```

“lsjobs” コマンドは、各 mugyu クライアントで走っているジョブを表示する。各ホストごとに、そのホストで実行されているジョブが表示される。各ジョブに関する情報は1行ごとに表示される。各項目はコンマで区切られる。

“user” の項目でどのユーザがジョブの投入したかを示す。“mugyu\_account” は、mugyu システムのアカウント名を示す。“mail\_address” はジョブが終了したことを通知する先のメールアドレスである。“jobID” は投入したジョブの ID 番号を示す。“pid” は mugyu クライアント上でのジョブのプロセス ID を示す。“submit\_time” はジョブが投入された時刻を示す。“from\_host” はジョブがどのホストから投入されたかを示す。“stdoutFileName” はジョブの標準出力がどのファイルに出力されるかを示す。“stderrFileName” はジョブの標準エラー出力がどのファイルに出力されるかを示す。“command” は、ジョブのプログラム名とその引数を示す。

## 実行中のジョブの強制終了

“bkill” コマンドを用いることで、実行中のジョブを強制終了することができる。“bkill” コマンドの第1引数には、強制終了させたいジョブが走っているホスト名を指定する。第2引数には強制終了させたいジョブの Job ID を指定する。

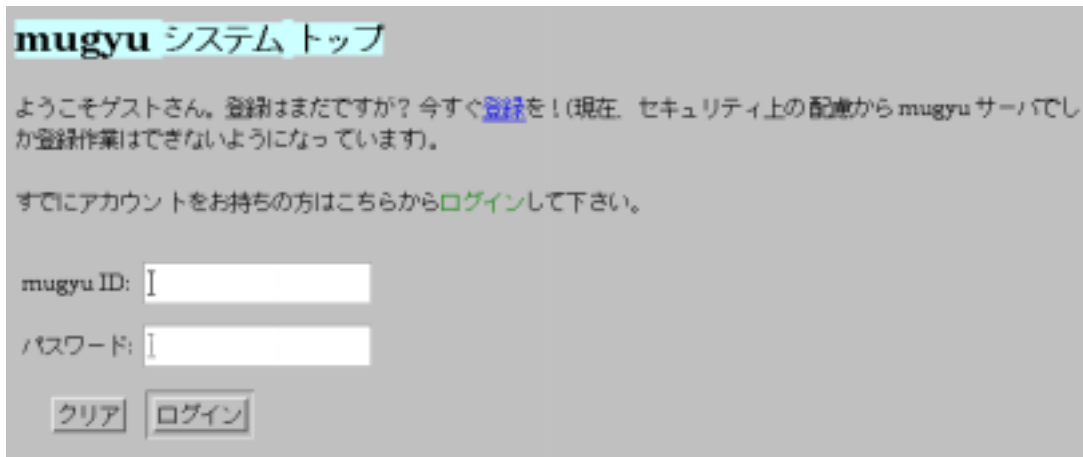


図 5.2: ウェブページによる mugyu システムへのログイン画面

## その他

クライアントでは inetd(8) スーパーデーモンが起動しており、telnet(1)、rlogin(1)、rsh(1) コマンドが使えるようになっている。“bsub” コマンドなどを使用せずに、1. rlogin コマンドでクライアントにログインする、2. バックグラウンドジョブとしてジョブを実行する、3. ログアウトする、というようにも利用することができる。

### 5.4.3 ウェブインターフェース

mugyu システムの 3 番目の層である、ウェブを用いたシステムの利用における機能を説明する。

#### ログイン画面

ウェブインターフェースを介して mugyu システムを利用するにはまず、図 5.2 に示したログイン画面から mugyu システムにログインする必要がある。ここでは mugyu ID(いわゆるログイン名にあたる)とパスワードを入力し、「ログイン」ボタンを押してログインすることができる。

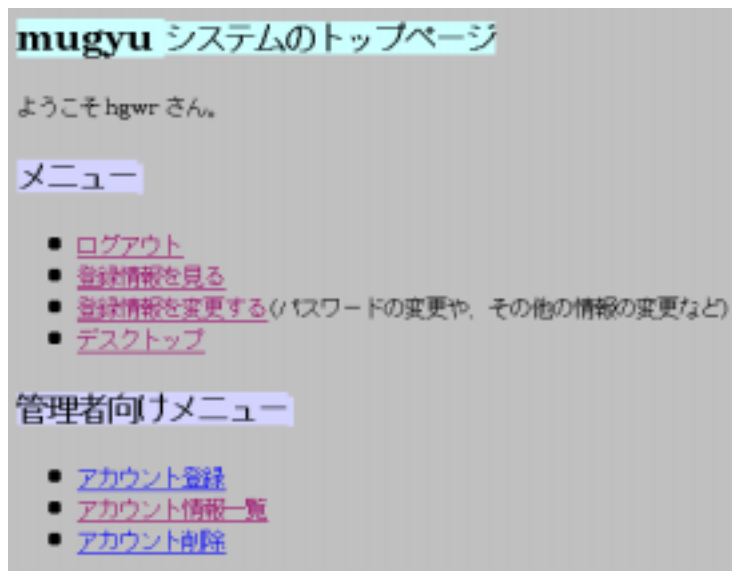


図 5.3: mugyu システムのトップページ画面

## トップページ画面

ログインに成功すると、図 5.3 に示した mugyu システムのトップページが表示される。このページからできることは次のようなことである。

- mugyu システムからログアウト
- 自分の mugyu システムへの登録情報を閲覧したり、変更するためのページへ行く
- デスクトップ画面へ行く

ユーザが管理者グループに加入している場合には、このページから他のユーザの登録情報の閲覧などができる。

## デスクトップ画面

ウェブインターフェースから mugyu システムを使用する際に、最も重要なのはデスクトップ画面(図 5.4)である。この画面は大きく分けて二つの部分からなる。図 5.4 の中ほどで区切られた線の上側には、操作を行った際にシステムから送られてくるメッセージが表示される。下側に表示されるのは、現在作業を行っているディレクトリに関する



情報とメニューバーである。ディレクトリに存在するそれぞれのファイルは、操作の対象としてそれぞれ選択できるようになっている。また、メニューバーをクリックすると、いくつかの操作名が表示され、そのうちどれかを選択し「実行」ボタンを押すことで希望する操作を行えるようになっている。

デスクトップ画面から出来ることは、大きく分けて3つある。ファイルに関する操作、ファイルのコンパイルや実行に関する操作、ジョブを投入するなどのジョブに関する操作である。以下では、この3つの操作について、それぞれ詳しく紹介する。

ファイルに対する操作は次のようなことができる。

- ファイルの内容の表示
- ファイルの内容の編集
- ファイルの新規作成
- ファイルの削除
- ファイルの名前の変更
- ファイルのアップロード
- ファイルのダウンロード
- ディレクトリの作成
- ディレクトリの削除
- ディレクトリの名前の変更
- 現在作業中のディレクトリの変更
- ファイルのコピー
- ファイルの移動

ファイルの実行などに関する操作は次のようなことができる。1. C 言語で書かれたファイルをコンパイルする。2. プログラムを保守、更新、再作成するプログラム `make(1)` コマンドの実行。3. デスクトップ画面で選択したファイルを実行する。

ジョブに関する操作は次のようなことができる。

- ジョブの投入
- 実行中のジョブの強制終了
- 実行中のジョブの一覧
- 今まで投入したジョブの履歴
- 今まで投入したジョブの履歴のクリア

### 操作の例: ジョブの投入

ウェブインターフェースを利用してジョブを投入する方法を説明する。まず、デスクトップ画面で、投入したいプログラムファイルを選択する。次に、「ジョブ」と表示されているメニューバーをクリックしメニューを開き、「ジョブ投入」を選択する。そして「実行」ボタンを押す。すると、デスクトップ画面の「メッセージ」の部分に、投入するプログラムへのコマンドライン引数を求める表示がなされる (図 5.5)。ここでコマンドライン引数を設定し、「ジョブ投入」ボタンを押すとジョブが投入される。

### 操作の例: 実行中のジョブの一覧の表示

現在実行中のジョブの一覧を見るには、まず、「ジョブ」と表示されているメニューバーをクリックしメニューを開き、「実行中のジョブ」を選択する。そして「実行」ボタンを押す。すると、デスクトップ画面の「メッセージ」の部分に、現在実行中のジョブの一覧が表示される (図 5.6)。ジョブに関する情報で表示されるのは、ジョブ ID、プロセス ID、投入先のホスト名 (すなわちジョブが実行されているホスト名)、投入した時刻、ジョブの標準出力への出力が保存されるファイルの名前、ジョブの標準エラー出力への出力が保存されるファイルの名前、ジョブとして実行されているコマンドとそのコマンドライン引数である。



図 5.4: mugyu システムへのデスクトップページ画面。左下のメニューバーを開くと、利用できる操作がメニュー一覧となって表示される。右したには現在のディレクトリに存在するファイルおよびサブディレクトリが表示される。それぞれのファイルおよびディレクトリは、ボタンによって選択可能になっている。

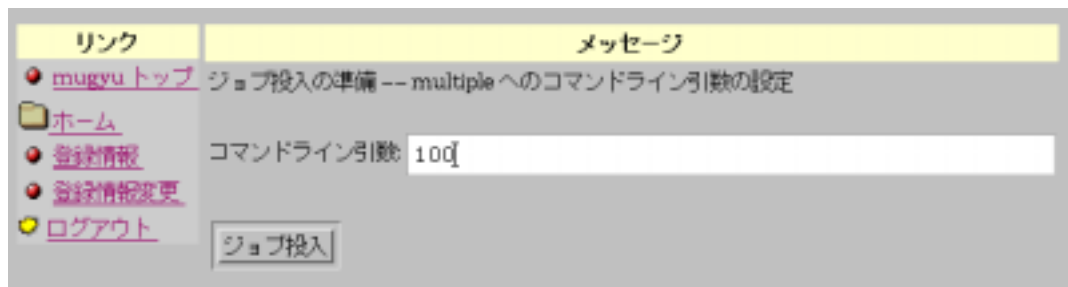


図 5.5: ウェブインターフェースを利用したジョブの投入。選択したコマンドへの引数を設定し、「ジョブ投入」ボタンを押すとジョブが投入される。

| リンク      | メッセージ     |           |                          |                                                               |  |
|----------|-----------|-----------|--------------------------|---------------------------------------------------------------|--|
| ● mugyu  | 実行中のジョブ一覧 |           |                          |                                                               |  |
| ● トップ    |           |           |                          |                                                               |  |
| ● ホーム    |           |           |                          |                                                               |  |
| ● 登録情報   |           |           |                          |                                                               |  |
| ● 登録情報変更 |           |           |                          |                                                               |  |
| ● ログアウト  |           |           |                          |                                                               |  |
| ジョブ ID   | pid       | 投入先ホスト    | 投入時刻                     | 標準出力ファイル                                                      |  |
| 4        | 498       | kspc99h03 | Sat Jan 15 08:49:32 2000 | hgwr/src/multiple/mugyu_kspc99h03_3_2000_0115_064932.stdout h |  |

図 5.6: ウェブから投入したジョブの一覧を見る。ジョブに関する情報は多数あるので画面から右の部分が切れてしまっている。しかし、ウェブブラウザによるスクロールで見ることができる。

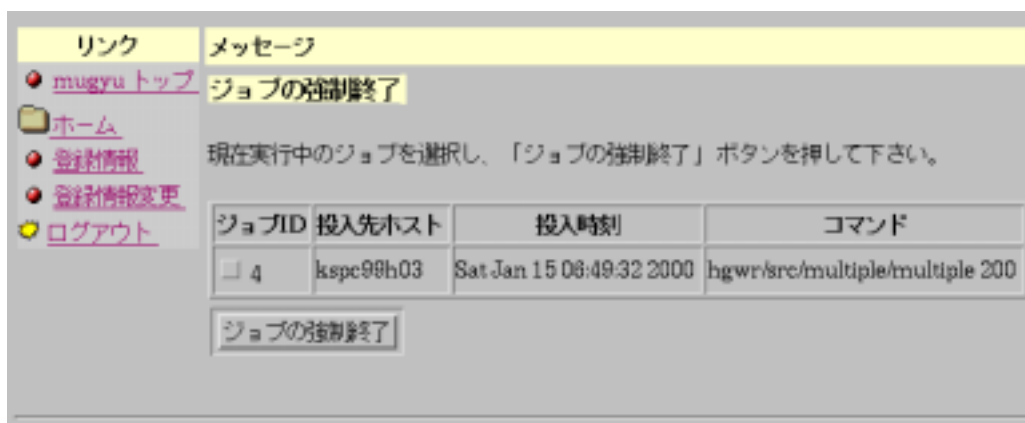


図 5.7: 投入したジョブをウェブインターフェースを利用して強制終了する。強制終了させたいジョブを選択し、「強制終了」ボタンを押すと選択したジョブが強制終了される。

### 操作の例: 実行中のジョブの強制終了

現在実行中のジョブを強制終了させるには、まず、「ジョブ」と表示されているメニューバーをクリックしメニューを開き、「ジョブの強制終了」を選択する。そして「実行」ボタンを押す。すると、デスクトップ画面の「メッセージ」の部分に、現在実行中のジョブの一覧が簡易表示される (図 5.7)。

ここで、強制終了させたいジョブを選択し、「ジョブの強制終了」ボタンを押すと選択したジョブが強制終了される。

## 5.5 システムの仕組み

### 5.5.1 ブート/ルートフロッピーの構成

クライアントはフロッピーで立ち上げなくてはならない。このためフロッピーに入りきらないものは、起動後に NFS でサーバのディスクがアクセスできるようになってから参照するようにした。

ブートフロッピーには、ブートローダと Linux のカーネルが入っている。カーネルは、mugyu クライアントにさせる PC の LAN カードを認識でき、RAM ディスクをサポートしている必要がある。

ルートフロッピーには、圧縮したルートファイルシステムが入っている。

前述の通り、フロッピーに入りきらない共有ライブラリを読み込んだり、クライアント上で実行するプログラムの結果を保存するためには NFS でサーバのディレクトリをマウントする必要がある。したがって、特にルートファイルシステムのなかで重要なものは、NFS を実現するための一連のプログラム (inetd(8), rpc.portmap, rpc.mountd, rpc.nfsd など<sup>1</sup>) である。

これらの作成に当たっては、Slackware 3.6 (<http://www.slackware.org/>) のブートディスク、レスキューディスクおよび文献 [6] を参考にした。

### 5.5.2 クライアントの設定時の GUI

curses(3) ライブラリを利用した Savio Lam による dialog コマンドを利用した。

### 5.5.3 クライアントの起動時に行われること

ブートディスクをフロッピーディスクドライブへ入れて PC を再起動すると、ブートフロッピー中のカーネルが読み込まれ、デバイスが認識される。

ブートフロッピーの読み込みが終了すると、カーネルがルートフロッピーを入れよ、との指示を出す。その指示に従い、ルートフロッピーをフロッピーディスクドライブへ挿入しリターンを押すと、圧縮されたルートファイルシステムが読み込まれる。この時点で、システムはルートファイルシステム (の /bin または /sbin ディレクトリ) で init(8) プログラムを見つけ、これを実行する。init(8) プログラムは設定ファイル /etc/inittab を読み込み、“sysinit” や “boot” や “bootwait” と書いてある行を探し、そこで指定されているコマンドを実行する。mugyu システムでは、システム起動スクリプトは /etc/rc スクリプトである。

最初に /etc/rc スクリプトは /etc/fstab ファイルに記述された NFS 以外のディスクのマウントをする。つぎに、/etc/rc.d/rc.variable\_settings スクリプトを実行し、第 5.4.1 節にあげた各項目について、ユーザに設定を求める。ユーザによる設定が終ると、それぞれの項目は対応するシェル変数に保存され後に使用されることになる。

---

<sup>1</sup>システムによって名前が異なる。

次に、`/etc/rc` スクリプトはユーザによる設定内容を用いて、`/etc/template` ディレクトリ以下にあるシステムの設定ファイルのテンプレートの穴を埋めて、`/etc/hostname` ファイルや `/etc/hosts` ファイルなどを作成し更新する。

さらに、`/etc/rc` スクリプトは `/etc/rc.d/rc.inet1` スクリプトを実行する。`/etc/rc.d/rc.inet1` スクリプトはネットワークシステムの基本となる部分の設定を行う。具体的にはループバックインターフェースや、イーサネットインターフェース、ゲイトウェイ経由の経路制御などの設定を行う。

最後に `/etc/rc` スクリプトは `/etc/rc.d/rc.inet2` スクリプトを実行する。`/etc/rc.d/rc.inet2` スクリプトは、RPC ポートマップの起動、NFS 経由のディスクのマウント、NFS 経由のディスクに存在する共有ライブラリへのキャッシュの更新、NFS サーバと時刻を同期させるための `xntpd` デーモンの起動、`inetd(8)` スーパーデーモンの起動、NIS の設定、最後に `mugyumentry` コマンドを実行し、`mugyud` デーモンプログラムを実行する。

`mugyumentry` コマンドは `mugyu` サーバの特定のディレクトリに自身のホスト名の名前のついたファイルを作成する。このことにより、`mugyu` サーバはシステムで利用できる `mugyu` クライアントを把握することができる。

`/etc/rc` スクリプトが終了すると制御は `init(8)` プログラムに戻る。`init(8)` プログラムは続いてデフォルトのランレベルに入る。実行レベルの行では `agetty(8)` プログラムが指定されている。`agetty(8)` プログラムはコンソールや `tty` 経由の通信を処理する。`agetty(8)` プログラムは “login:” というプロンプトを出す。`agetty` プログラムは続いて `login(1)` プログラムを実行し、これがログイン認証やユーザセッションの設定を行う。このように `/etc/rc` スクリプト以後の処理は通常の UNIX 的なシステムと変わらない。

#### 5.5.4 各コマンドの仕組み

この節に紹介した各プログラムの作成に当たっては、文献 [5]、[7]、[27]、[29]、[31] および [28] を参考にした。

### ジョブの投入

ユーザはジョブ投入プログラム `bsub` を利用してジョブの投入を行う。

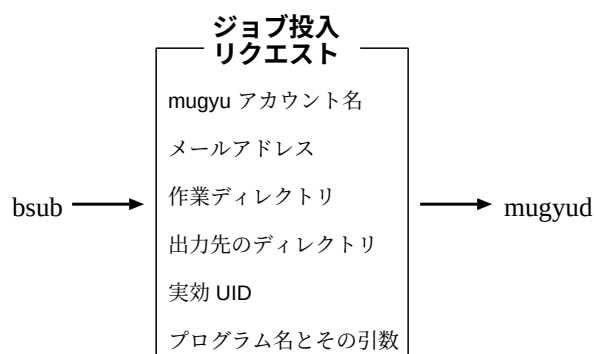


図 5.8: bsub から mugyud へのリクエストの送付。図のように bsub プログラムから mugyud へのジョブ投入リクエストは、実行するプログラム名の他にもさまざまなデータを含む。

mugyu クライアントは起動時に、NFS 経由で mugyu サーバの特定のディレクトリに自ホストの名前と同名のファイルを作成する。このため bsub プログラム他、mugyu システムのそれぞれのプログラムはジョブの投入可能なホスト名のリストを知ることができる。

まず bsub プログラムは、リストにある各ホストの mugyud デーモンプログラムに“load” リクエストを送出する。“load” リクエストを受け取った mugyud デーモンプログラムは、自ホストの負荷の状態を返す。この仕組みによって bsub プログラムは負荷の低いホストを調べ、最も負荷の低いホストにジョブを投入することができる。

次に、bsub プログラムは、getuid(2) システムコールを利用してどのユーザの権限で自分が実行されたかを調べる。bsub プログラムはそのユーザ権限で、指定されたプログラムを実行するように mugyud デーモンプログラムへリクエストを送る (図 5.8)。mugyud デーモンプログラムへ送付する内容は、ジョブの持ち主の mugyu アカウント名、ジョブが終了したことを通知する先のメールアドレス、作業ディレクトリ、出力先のディレクトリ、誰の権限でプログラムを実行するか、そして実行するプログラム名とその引数、である。

さらに、リクエストを受け取った mugyud デーモンプログラムは fork(2) システムコールを実行する。mugyud デーモンプログラムは root 権限で実行されている。そのため ジョブとして実行されることになる子プロセスを、setuid(2) システムコールを呼ぶことで、bsub プログラムから指定されたユーザ権限にする。

ここでプログラムの実効 UID がセキュリティ上の問題になる。bsub プログラムは、



mugyud デーモンプログラムに bsub プログラムの実行者の権限でプログラムの実行を要求するので問題ない。しかし、悪意のあるユーザが bsub プログラムから mugyud デーモンプログラムへのプロトコルをまねることができる。すなわち、悪意あるユーザが root 権限でプログラムを実行することを、mugyud デーモンプログラムに要求する恐れが出てくる。これを防ぐために、mugyud デーモンプログラムは、要求されたユーザ権限でプログラムを実行しても良いかどうか調べる機能を持っている。

ジョブを投入したいユーザは、あらかじめ mugyu システム独自のパスワードファイルに、ユーザ名と UID とパスワードを登録しておかなければならない<sup>2</sup>。さらに、ユーザはホームディレクトリに “.mugyurc” ファイルを準備しなくてはならない。ユーザは “.mugyurc” ファイルに、mugyu システムのパスワードファイルにある暗号化されたパスワードのもとになっている、平文のパスワードを書き込んでおく必要がある。なおかつ “.mugyurc” ファイルを、ユーザ自身以外には誰にもみられないように、ファイルのモードを設定しておく必要がある。

mugyud デーモンプログラムは次に示す調査項目を順に調べ、条件に合わなかった時点で要求されたプログラムの実行を中止する。

1. mugyu システムのパスワードファイルに、要求された権限のユーザ ID の登録があるか。
2. 要求された権限のユーザのホームディレクトリに “.mugyurc” ファイルがあるか。
3. “.mugyurc” ファイルの持ち主が要求された権限のユーザか。
4. “.mugyurc” がレギュラーファイルか (シンボリックリンクなどでないか)。
5. “.mugyurc” のファイルのモードをそのユーザのみ読めるように制限してあるか。
6. mugyu システムのパスワードファイルにある暗号化されたパスワードと、 “.mugyurc” ファイルの内容を暗号化したものは同じ文字列か。

このような仕組みで、悪意あるユーザの不正な権限でのプログラム実行を防いでいる。ユーザのチェックを行った後、リクエストとして受け取ったプログラム名を exec(2) システムコールを用いて実行する。プログラムの標準出力およびエラー出力は、特定のファイルへ結び付けられる。

---

<sup>2</sup>mugyu システム独自のパスワードファイルを保守、更新、変更するためのコマンドを用意した。mugyuuseradd と mugyupasswd である。

このような仕組みは rsh(1) コマンドと比較して次のような利点がある。rsh(1) コマンドはリモートホスト上で実行されているプログラムの標準出力とエラー出力を端末に接続するので、リモートホスト上で実行しているプログラムが終了するまで、ローカルホスト上で rsh(1) プログラムの実行が終わらない。しかし、上の方式だとローカルホスト上で実行される処理は、リモートホストの mugyud デーモンプログラムへ実行するプログラム名などを送信するだけである。

## 実行中のジョブの閲覧

実行中のジョブの閲覧をするには “lsjobs” プログラムを用いる。各 mugyu クライアント上で実行中の mugyud デーモンプログラムへ、実行中のジョブ一覧を返すようにリクエストを送る、というように実装されている。

mugyud デーモンプログラムはマルチスレッドプログラムである。リクエストを受けると新しくスレッドを生成する。ジョブを受け付けて fork(2) システムコールおよび exec(2) システムコールの実行をするのは、新しく生成された子スレッドである。親スレッドは子スレッドを生成したあと、またすぐにリクエスト待ちに入る。子スレッドは、fork(2) システムコールを実行した子プロセスに関する情報を、mugyd が保持する「実行中のジョブリスト」に加え、wait(2) システムコールを実行する。子プロセスが終了すると、その子プロセスをジョブリストから削除し、子スレッドは消滅する。このような仕組みで mugyud デーモンプログラムはジョブリストを更新し、同じホスト上で実行中のすべてのジョブを常に把握している。

### 5.5.5 ウェブインターフェース

mugyu システムへログインする際にパスワードを送信することから、HTTP サーバとの通信は SSL (Secure Socket Layer) による HTTP、いわゆる https を介して行うようにした。これを実現するため、HTTP サーバーには apache\_1.3.9+ssl\_1.3.7 パッケージを使用した。

ウェブページの作成には、php-3.0.12jp-beta4 パッケージを使用した。

ウェブインターフェースによる mugyu システムのユーザ情報管理、ジョブの履歴の管

理などには RDBMS(Relational Database Management System)である postgresql-6.5.3 を使用した。

ウェブインターフェースにおけるユーザ認証にはクッキーを利用した。クッキーについては文献 [18] および [24] を参考にした。

# 謝辞

この研究にご協力いただいた多くの方々に感謝の意を表します。遺伝子知識システム論講座のみなさまには、たくさんの貴重な助言をいただきました。知識科学教育研究センターの山下邦宏助教授には、こころよく電算室の数十台のPCの利用をお許しいただきました。遺伝子知識システム論講座の佐藤賢二助教授には、貴重な時間を私のために割いてアドバイスを下さいましたことに感謝致します。最後に、私の研究が小長谷明彦教授に多くの恩恵をこうむっていることを明記し、教授の好意と寛容に謝意を表したいと思います。

## 参考文献

- [1] 安居院猛, 長尾智晴. ジェネティックアルゴリズム. 昭晃堂, 1993.
- [2] Bruce Alberts, Dennis Bray, Julian Lewis, Martin Raff, Keith Roberts, and James D. Watson. *Molecular Biology of The Cell*. Garland Publishing, Inc., 717 Fifth Avenue, New Yourk, NY 10022, 3rd edition, 1994. (邦訳: 「細胞の分子生物学第3版」中村圭子、藤山秋佐夫、松原謙一監訳、教育社、1995年).
- [3] Masanori Arita. Simfly2: Simulation of a fly embryo. In *Genome Informatics 6*, pp. 29–38, 1995. Genome Informatics Workshop 1995, Oral Session.
- [4] Masanori Arita, Masami Hagiya, and Tomoki Shiratori. Simfly: A simulator for protein interaction in fly embryo. In *Genome Informatics 5*, pp. 230–231, 1994.
- [5] David R. Butenhof. *Programming with POSIX Threads*. Addison Wesley Longman, 1997. (邦訳: 「POSIX スレッドプログラミング」油井尊一訳、アジソン・ウェスレイ、1998).
- [6] Tom Fawcett (fawcett@croftj.net). The linux bootdisk howto, May 1999. (邦訳: 「The Linux Bootdisk HOWTO」藤原輝嘉、中野武雄訳、<http://www.linux.or.jp/JF/JFdocs/Bootdisk-HOWTO.html>, Jun 1999).
- [7] Enrich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley Publishing Company, 1995. (邦訳: 「オブジェクト指向における再利用のためのデザインパターン」本位田真一、吉田和樹訳、ソフトバンク、1998).

- [8] S. F. Gilbert. *Developmental Biology*, chapter 14. The Genetics of axis specification in *Drosophila*. Sinauer Associates Inc., 5th edition, 1997.
- [9] Shugo Hamahashi and Hiroaki Kitano. Simulation of *Drosophila* embryogenesis. In Christoph Adami [et.al.], editor, *Artificial Life VI: proceedings of the sixth International conference on Artificial Life*, pp. 151–160, 1998.
- [10] Volker Hartenstein. Overview: The segmentation genes of *Drosophila*, September 1998. adapted from “Atlas of *Drosophila* Development”, 1993 (by Volker Hartenstein), <http://zygote.swarthmore.edu/segmentationgenes1.html>.
- [11] Michael Hoch and Herbert Jäckle. Transcriptional regulation and spatial patterning in *Drosophila*. *Current Opinion in Genetics and Development*, Vol. 3, pp. 566–573, 1993.
- [12] 伊庭斉志. 遺伝的アルゴリズムの基礎. オーム社, 1994.
- [13] 伊庭斉志. 遺伝的プログラミング. 情報科学セミナー. 東京電気大学出版局, 1996.
- [14] 伊庭斉志. 進化的計算の方法. 東京大学出版会, 1998.
- [15] 北野宏明 (編) . 遺伝的アルゴリズム. 産業図書, 1993.
- [16] John R. Koza. *Genetic Programming: on the programming of computers by means of natural selection*. The MIT Press, Cambridge, Massachusetts, 1992.
- [17] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Subprograms*. The MIT Press, Cambridge, Massachusetts, 1994.
- [18] D. Kristol and L. Montulli. Http state management mechanism, 1997. Network Working Group, RFC 2109.
- [19] Michael Mahowald. Bicoid-dependent activation of posterior genes in *Drosophila* embryogenesis, Fall 1998. Developmental Genetics Seminar, Swarthmore College, <http://zygote.swarthmore.edu/hairystripe.html>.

- [20] David J. Montana. BBN Technical Report #7866 “Strongly Typed Genetic Programming”. Technical report, Bolt Beranek and Newman, Inc., 10 Moulton Street, Cambridge, MA 02138, May 1993.
- [21] Michael J. Pankratz and Herbert Jäckle. Making stripes in the *Drosophila* embryo. *Trends in Genetics*, Vol. 6, No. 9, pp. 287–293, 1990.
- [22] John Reinitz and David H. Sharp. Mechanism of *eve* stripe formation. *Mechanisms of Development*, Vol. 49, pp. 133–158, 1995.
- [23] Rolando Rivera-Pomar and Herbert Jäckle. From gradients to stripes in *Drosophila* embryogenesis: filling in the gaps. *Trends in Genetics*, Vol. 12, No. 11, pp. 478–483, 1996.
- [24] 佐藤正博. クッキーのしくみと CGI を使った利用例. オープンデザイン, No. 30, pp. 5–40, 2月号 1999.
- [25] Stephen Small, Rachel Kraut, Timothy Hoey, Rahul Warrior, and Micael Levine. Transcriptional regulation of a pair-rule stripe in *Drosophila*. *Genes & Development*, Vol. 5, pp. 827–839, 1991.
- [26] Stephen Small and Michael Levine. The initiation of pair-rule stripes in the *Drosophila* blastoderm. *Current Opinion in Genetics and Development*, Vol. 1, pp. 255–260, 1991.
- [27] W. Richard Stevens. *UNIX network programming*. Prentice Hall, 1990. (邦訳: 「UNIX ネットワークプログラミング」 篠田陽一訳、トッパン、1992 年).
- [28] 砂原秀樹, 石井秀治, 植原啓介, 林周志. プロフェッショナル BSD. アスキー, 1994.
- [29] Andrew S. Tanenbaum. *Distributed Operating Systems*. Prentice Hall, 1995. (邦訳: 「分散オペレーティングシステム」 水野忠則、鈴木健二、宮西洋太郎、佐藤文明訳、プレンティスホール、1996).

- [30] Christian Wolff, Melissa Pepling, Peter Gerger, and Martin Klingler. Structure and evolution of a pair-rule interaction element: *runt* regulatory sequences in *D. melanogaster* and *D. virilis*. *Mechanisms of Development*, Vol. 80, pp. 87–99, 1999.
- [31] 湯淺太一, 安村通晃, 中田登志之 (編) . bit 別冊 はじめての並列プログラミング. 共立出版, 6 月 1998 年.



# 付 録 A

## 実験 1

### A.1 実験 1 で得られた最良個体

実験 1 で得られたの最良個体を以下に示す。なお、“protein0” は bicoid、“protein1” は nanos、“protein9” は eve を示す。

```
(IF (> (ConcentrationOf protein9) 16.537628) protein9)
(IF (> (ConcentrationOf protein9) 25.586550) protein9)
(IF (> (ConcentrationOf protein9) 25.586550) protein9)
(IF (> (ConcentrationOf protein8) 18.024789) protein5)
(IF (> (ConcentrationOf protein8) 17.635735) protein1)
(IF (< (ConcentrationOf protein8) 10.136974) protein1)
(IF (> (ConcentrationOf protein9) 19.493214) protein9)
(IF (> (ConcentrationOf protein9) 19.493214) protein9)
(IF (> (ConcentrationOf protein9) 21.197850) protein9)
(IF (> (ConcentrationOf protein2) 45.907962) protein8)
(IF (< (ConcentrationOf protein0) 10.136974) protein3)
(IF (< (ConcentrationOf protein7) 13.387182) protein3)
(IF (> (ConcentrationOf protein8) 6.186284) protein2)
(IF (< (ConcentrationOf protein3) 25.340745) protein0)
(IF (> (ConcentrationOf protein2) 2.456680) protein0)
(IF (> (ConcentrationOf protein9) 16.537628) protein9)
```

(IF (< (ConcentrationOf protein1) 13.387182) protein9)  
(IF (> (ConcentrationOf protein9) 37.880912) protein9)  
(IF (> (ConcentrationOf protein2) 3.796834) protein8)  
(IF (> (ConcentrationOf protein3) 6.186284) protein9)  
(IF (> (ConcentrationOf protein9) 37.880912) protein9)  
(IF (> (ConcentrationOf protein4) 17.178871) protein0)  
(IF (> (ConcentrationOf protein9) 29.841667) protein6)  
(IF (> (ConcentrationOf protein4) 51.168421) protein2)

## A.2 最良個体の遺伝子調節タンパク質の分布の推移

実験1で得られた最良個体は、付録 A.1 に示した通りである。この個体を評価してもっとも正解に近づいたステップは 72 ステップ目であった。その時の各遺伝子調節タンパク質の分布を図 A.11 に示す。

以下では、「合成-拡散-削除」ステップごとの遺伝子調節タンパク質の分布の推移を一連の図を挙げて示す。

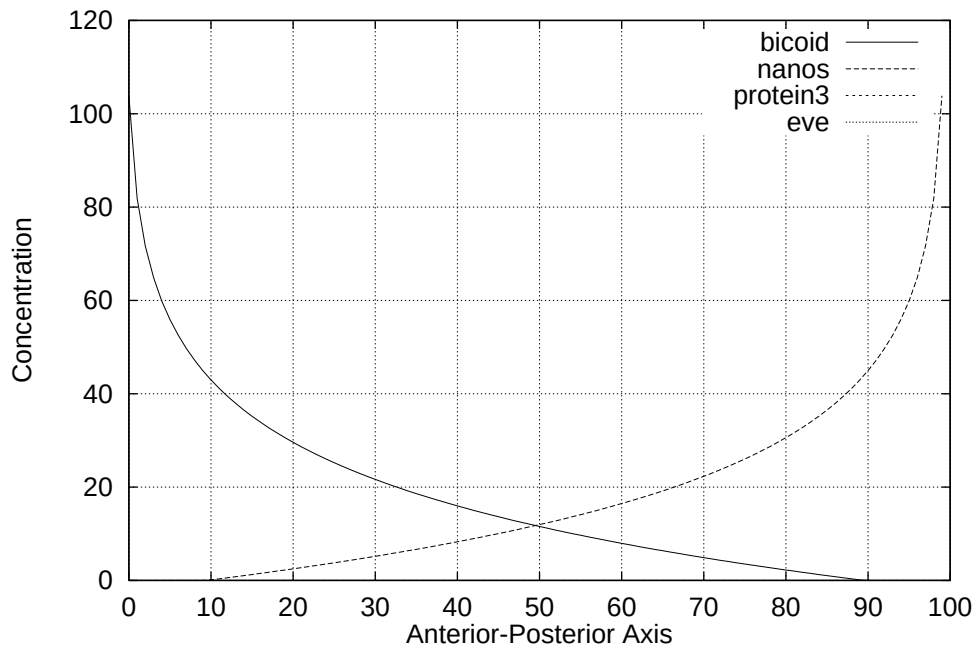


図 A.1: 実験 1 の「拡散-合成-削除」ステップ 0 の各遺伝子調節タンパク質の分布

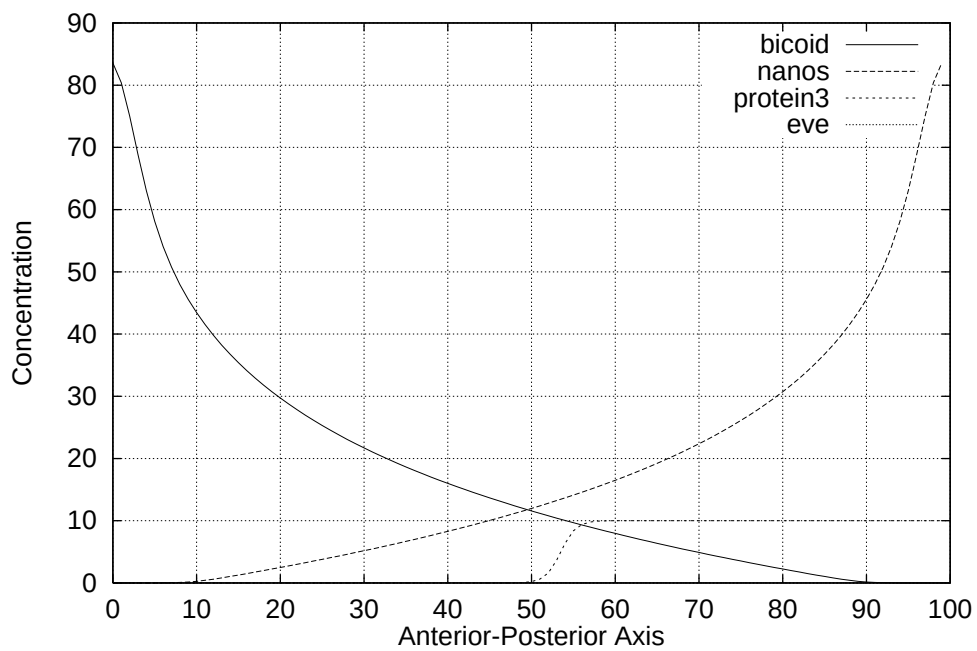


図 A.2: 実験 1 の「拡散-合成-削除」ステップ 10 の各遺伝子調節タンパク質の分布

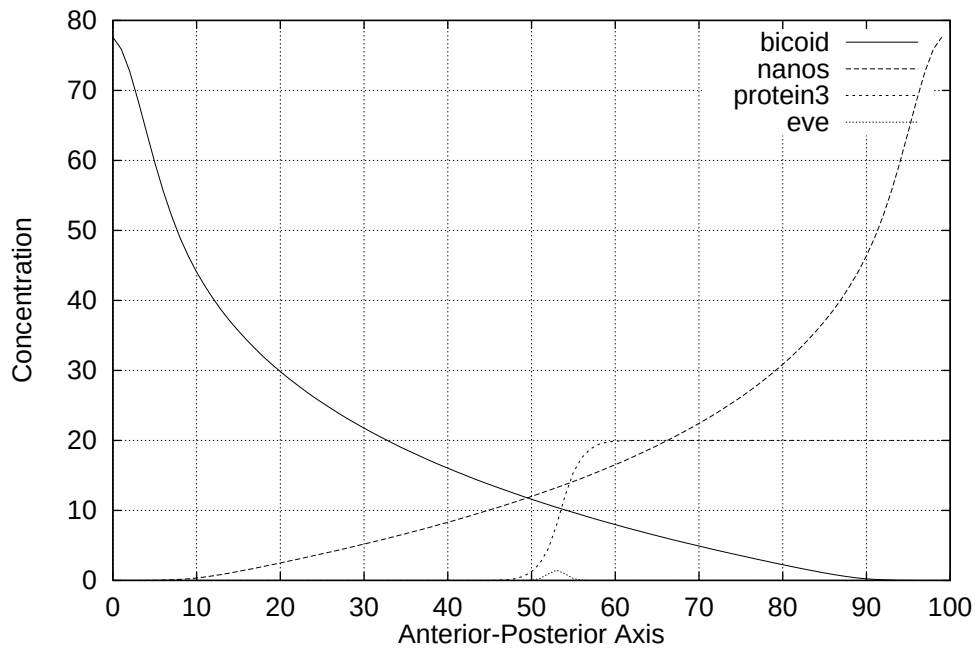


図 A.3: 実験 1 の「拡散-合成-削除」ステップ 20 の各遺伝子調節タンパク質の分布

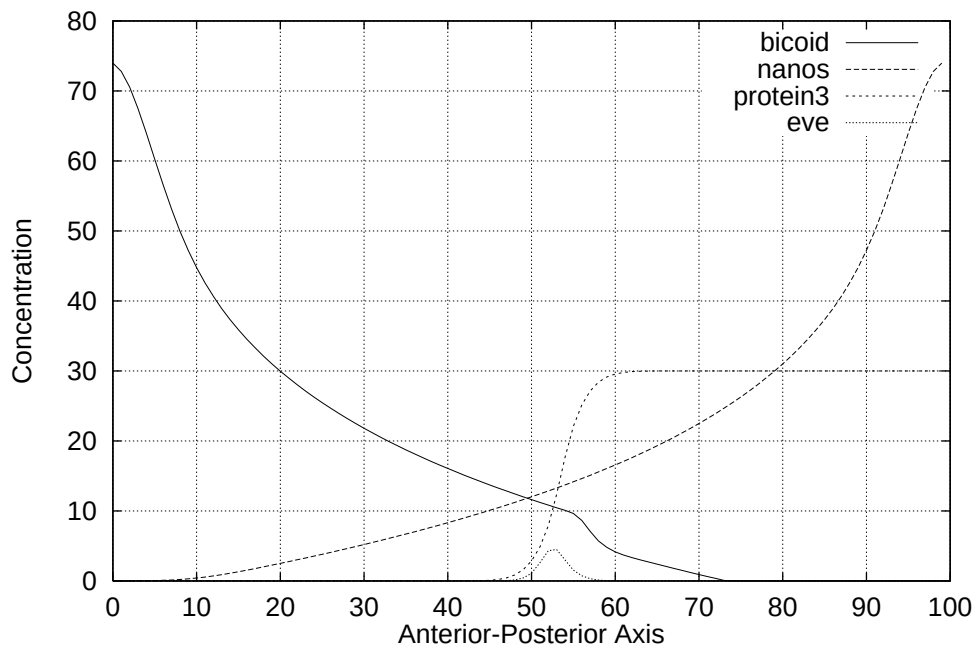


図 A.4: 実験 1 の「拡散-合成-削除」ステップ 30 の各遺伝子調節タンパク質の分布

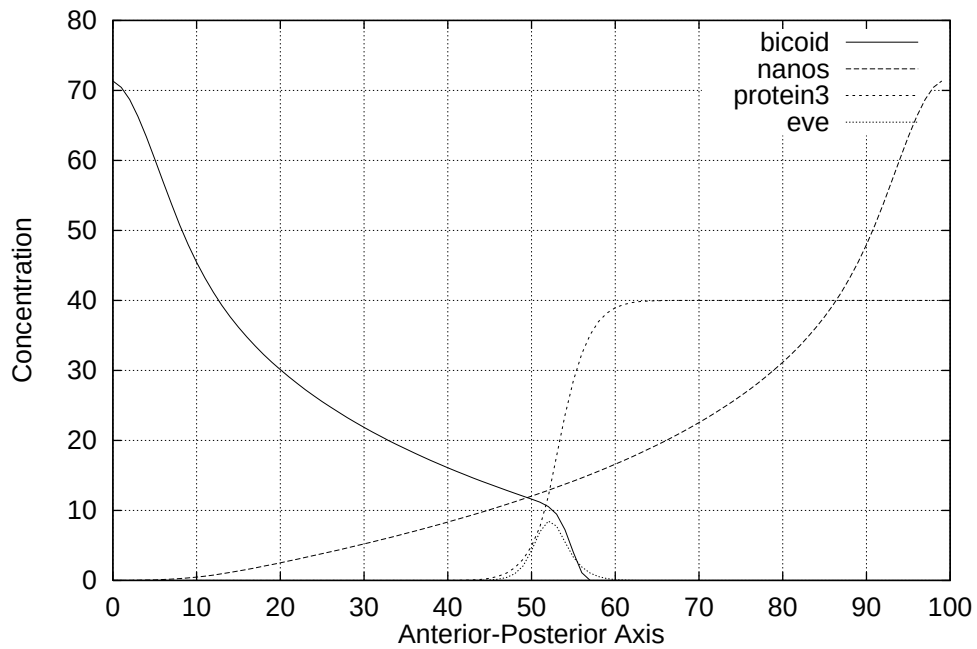


図 A.5: 実験 1 の「拡散-合成-削除」ステップ 40 の各遺伝子調節タンパク質の分布

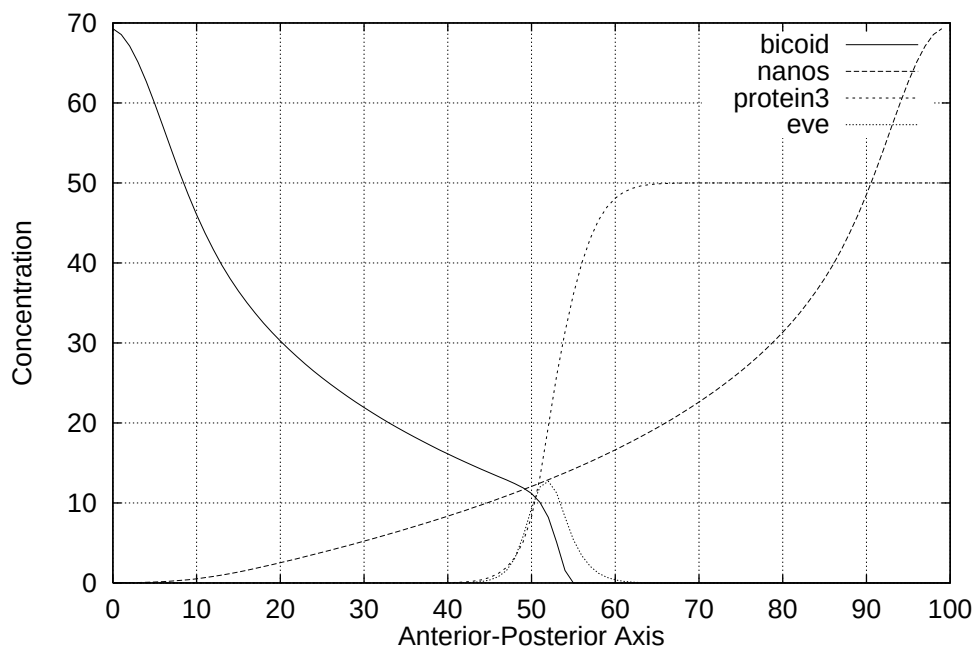


図 A.6: 実験 1 の「拡散-合成-削除」ステップ 50 の各遺伝子調節タンパク質の分布

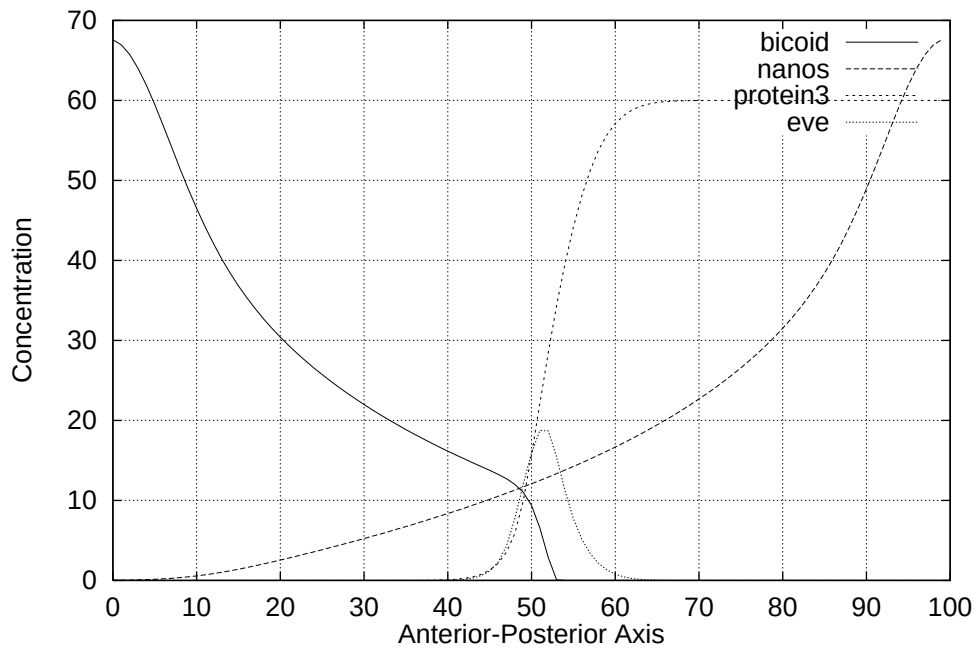


図 A.7: 実験 1 の「拡散-合成-削除」ステップ 60 の各遺伝子調節タンパク質の分布

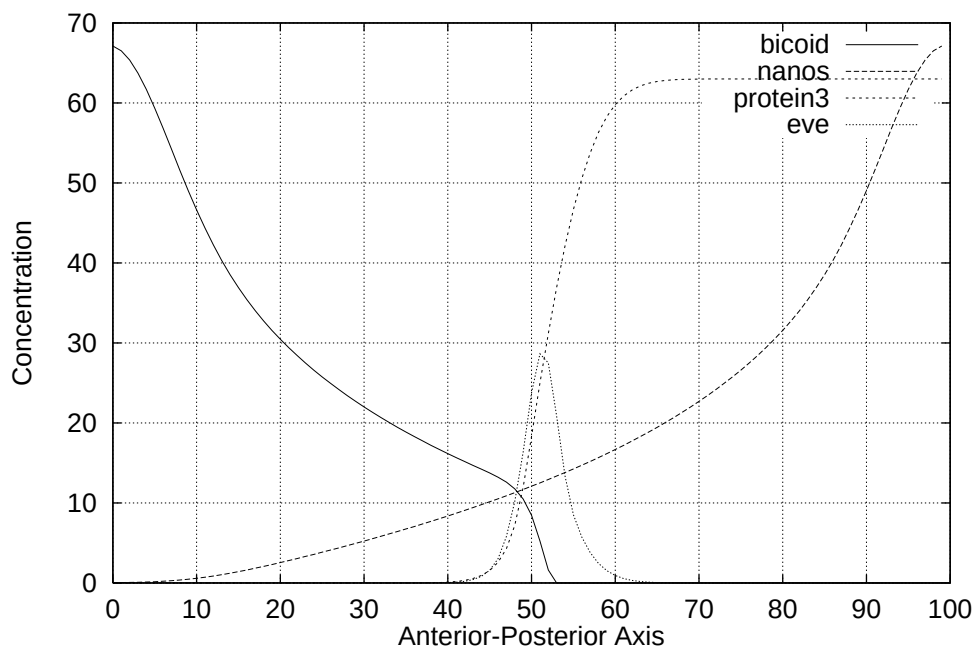


図 A.8: 実験 1 の「拡散-合成-削除」ステップ 63 の各遺伝子調節タンパク質の分布

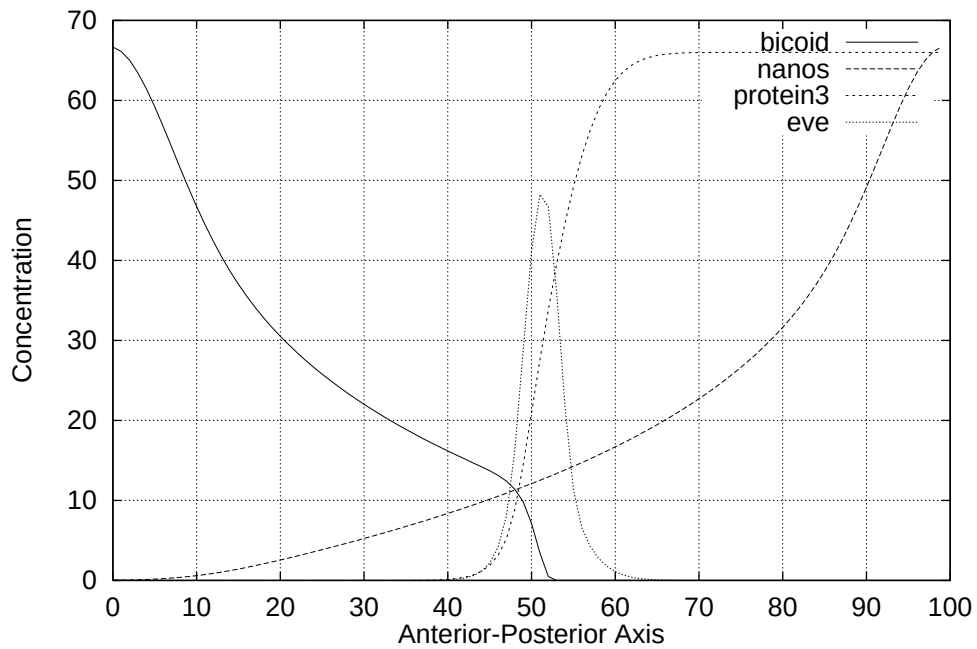


図 A.9: 実験 1 の「拡散-合成-削除」ステップ 66 の各遺伝子調節タンパク質の分布

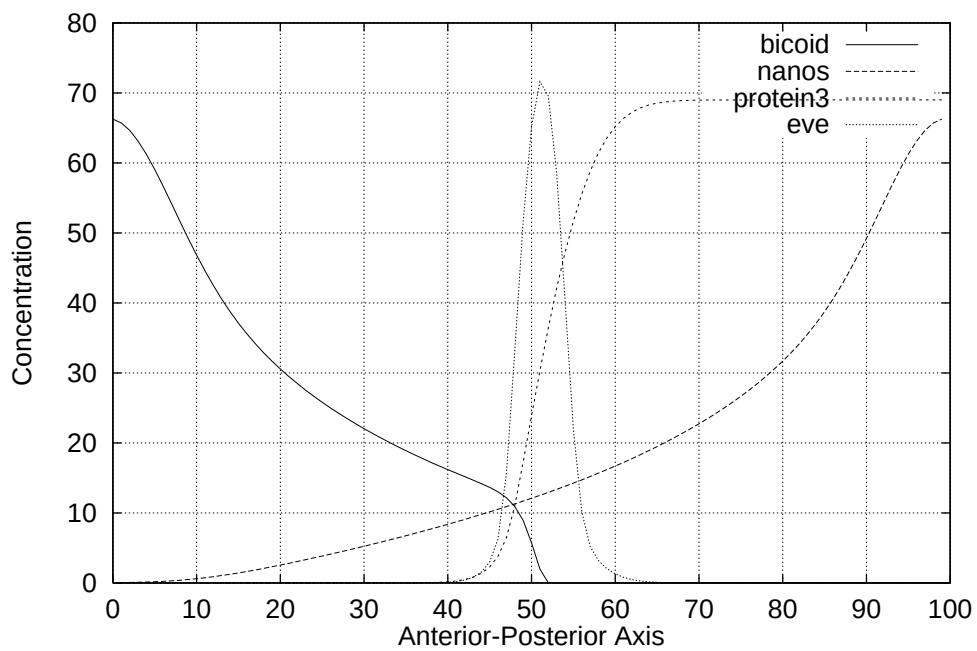


図 A.10: 実験 1 の「拡散-合成-削除」ステップ 69 の各遺伝子調節タンパク質の分布

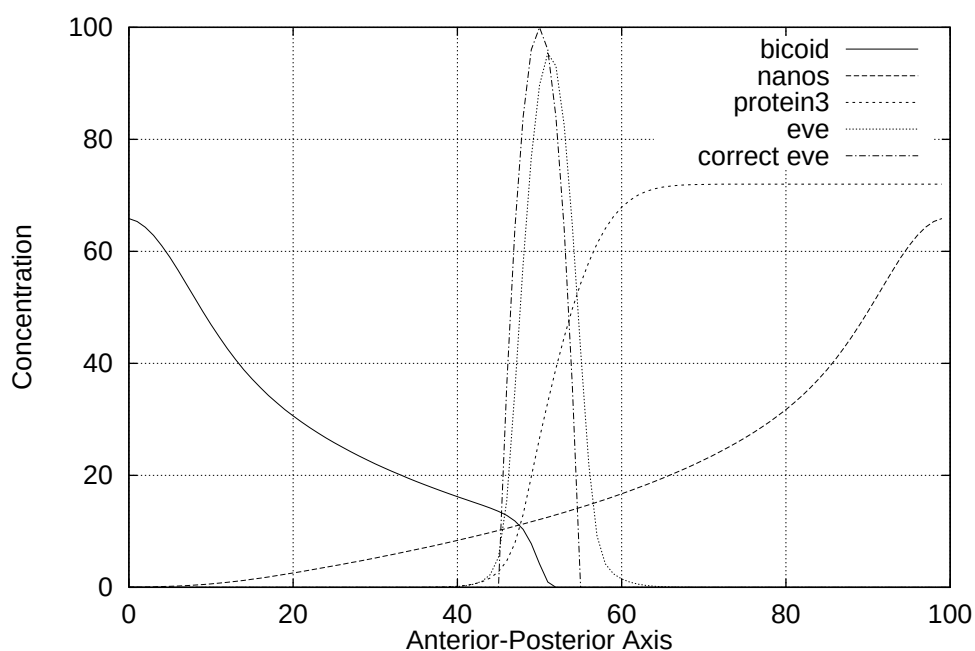


図 A.11: 実験1の「拡散-合成-削除」ステップ72の各遺伝子調節タンパク質の分布。このステップがもっとも正解と近かったときである。これ以後は正解から離れていく。

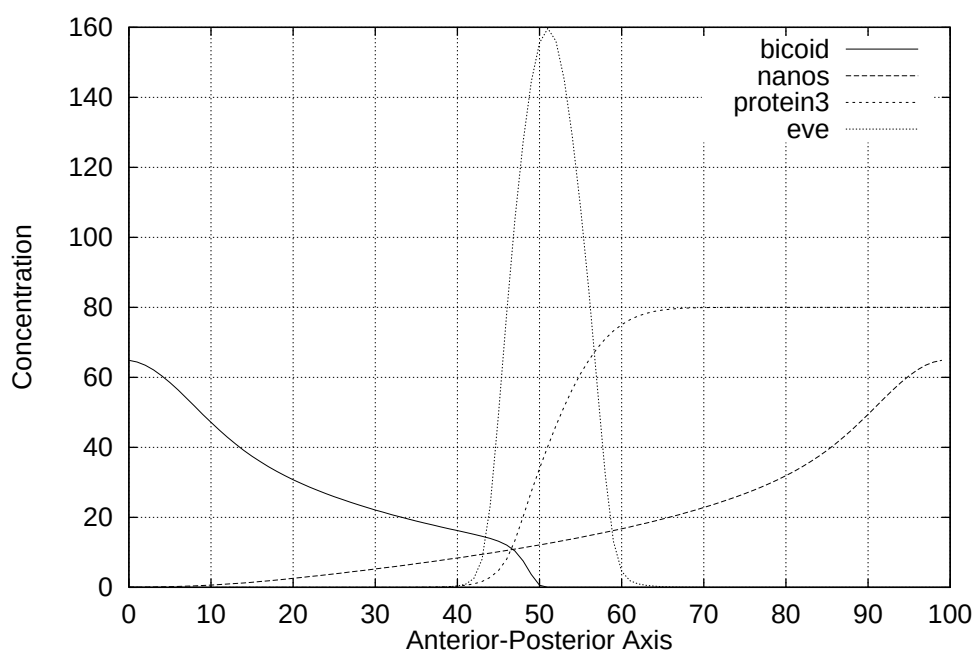


図 A.12: 実験1の「拡散-合成-削除」ステップ80の各遺伝子調節タンパク質の分布。



# 付 録 B

## 実験 2

### B.1 実験 2 で得られた最良個体

実験 2 で得られたの最良個体を以下に示す。なお、“protein0” は bicoid、“protein1” は nanos、“protein9” は eve を示す。

```
(IF (> (ConcentrationOf protein9) 51.952572) protein9)
(IF (> (ConcentrationOf protein9) 39.182861) protein9)
(IF (> (ConcentrationOf protein9) 39.182861) protein9)
(IF (< (ConcentrationOf protein0) 5.801844) protein9)
(IF (> (ConcentrationOf protein0) 2.172040) protein8)
(IF (< (ConcentrationOf protein0) 5.240812) protein0)
(IF (< (ConcentrationOf protein7) 96.180328) protein5)
(IF (< (ConcentrationOf protein0) 7.062997) protein7)
(IF (< (ConcentrationOf protein0) 7.062997) protein7)
(IF (> (ConcentrationOf protein9) 30.611979) protein9)
(IF (> (ConcentrationOf protein7) 0.953988) protein0)
(IF (< (ConcentrationOf protein0) 5.801844) protein9)
(IF (> (ConcentrationOf protein9) 20.594387) protein9)
(IF (> (ConcentrationOf protein9) 39.182861) protein9)
(IF (> (ConcentrationOf protein9) 17.702295) protein4)
(IF (AND (OR (OR (OR (NOT (NOT (AND (NOT (> (ConcentrationOf \
```

```

protein9) 39.182861)) (AND (> (ConcentrationOf protein9) 9.631942) \
(< (ConcentrationOf protein5) 10.862033)))) (OR (AND (OR (> \
(ConcentrationOf protein6) 34.774349) (NOT (> (ConcentrationOf \
protein3) 25.502499))) (AND (> (ConcentrationOf protein3) 0.854390) \
(OR (> (ConcentrationOf protein1) 4.647651) (< (ConcentrationOf \
protein0) 15.023929)))) (NOT (> (ConcentrationOf protein5) \
19.398758)))) (OR (NOT (NOT (AND (> (ConcentrationOf protein2) \
52.940987) (AND (NOT (< (ConcentrationOf protein5) 37.639664)) (< \
(ConcentrationOf protein2) 5.085065))))) (OR (< (ConcentrationOf \
protein2) 0.276982) (OR (< (ConcentrationOf protein7) 41.881824) \
(AND (< (ConcentrationOf protein3) 17.142811) (< (ConcentrationOf \
protein2) 83.039360)))))) (< (ConcentrationOf protein5) 2.229738)) \
(OR (OR (< (ConcentrationOf protein6) 4.796277) (> (ConcentrationOf \
protein0) 17.702295)) (> (ConcentrationOf protein3) 43.898571))) \
protein5)
(IF (< (ConcentrationOf protein8) 2.534261) protein5)
(IF (< (ConcentrationOf protein5) 2.534261) protein0)
(IF (< (ConcentrationOf protein6) 20.594387) protein1)
(IF (> (ConcentrationOf protein9) 30.611979) protein9)
(IF (> (ConcentrationOf protein9) 39.182861) protein9)
(IF (> (ConcentrationOf protein9) 51.952572) protein9)
(IF (> (ConcentrationOf protein9) 30.611979) protein9)
(IF (> (ConcentrationOf protein9) 51.952572) protein9)
(IF (> (ConcentrationOf protein9) 25.441944) protein9)
(IF (> (ConcentrationOf protein8) 25.441944) protein9)
(IF (< (ConcentrationOf protein0) 7.062997) protein4)
(IF (< (ConcentrationOf protein0) 2.534261) protein9)
(IF (< (ConcentrationOf protein4) 66.098541) protein1)
(IF (> (ConcentrationOf protein9) 0.364387) protein1)

```

```

(IF (> (ConcentrationOf protein5) 15.023929) protein7)
(IF (> (ConcentrationOf protein8) 0.364387) protein4)
(IF (> (ConcentrationOf protein5) 15.023929) protein7)
(IF (> (ConcentrationOf protein9) 39.182861) protein8)
(IF (> (ConcentrationOf protein5) 15.023929) protein3)
(IF (< (ConcentrationOf protein7) 41.013336) protein8)
(IF (< (ConcentrationOf protein0) 7.062997) protein2)
(IF (< (ConcentrationOf protein8) 16.934481) protein2)
(IF (< (ConcentrationOf protein7) 16.934481) protein2)
(IF (< (ConcentrationOf protein6) 2.663540) protein1)
(IF (< (ConcentrationOf protein5) 0.180849) protein7)
(IF (> (ConcentrationOf protein9) 39.182861) protein9)
(IF (> (ConcentrationOf protein9) 30.611979) protein9)
(IF (< (ConcentrationOf protein3) 13.232692) protein1)
(IF (< (ConcentrationOf protein2) 1.176075) protein1)
(IF (> (ConcentrationOf protein7) 5.285993) protein5)

```

## B.2 最良個体の遺伝子調節タンパク質の分布の推移

実験2で得られた最良個体は、付録 B.1 に示した通りである。この個体を評価してもっとも正解に近づいたステップは 42 ステップ目であった。その時の各遺伝子調節タンパク質の分布を図 B.10 に示す。

以下では、「合成-拡散-削除」ステップごとの遺伝子調節タンパク質の分布の推移を一連の図を挙げて示す。

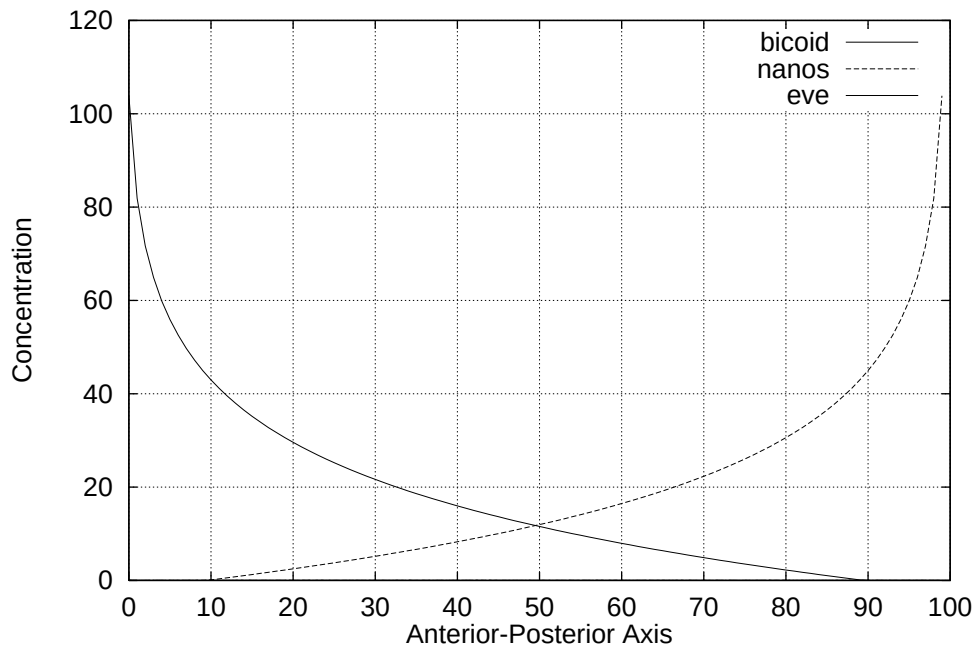


図 B.1: 実験 2 の「拡散-合成-削除」ステップ 00 の各遺伝子調節タンパク質の分布

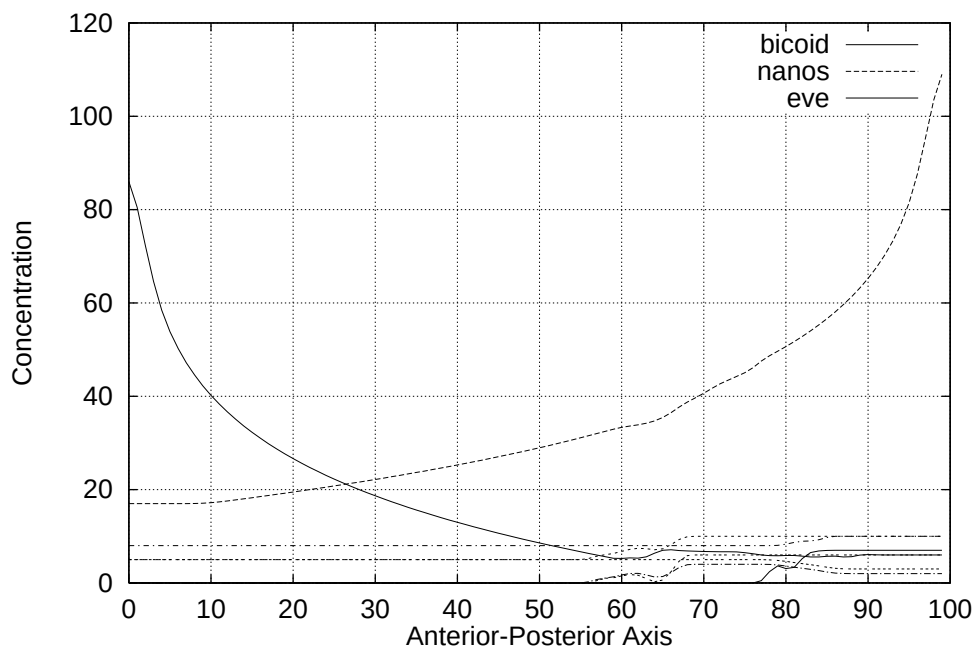


図 B.2: 実験 2 の「拡散-合成-削除」ステップ 05 の各遺伝子調節タンパク質の分布

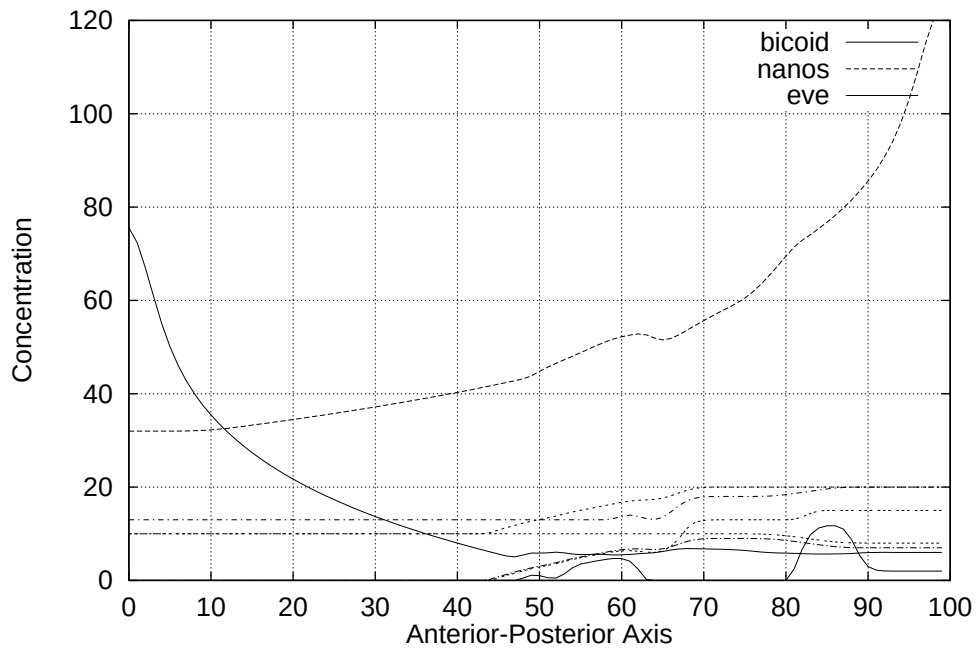


図 B.3: 実験 2 の「拡散-合成-削除」ステップ 10 の各遺伝子調節タンパク質の分布

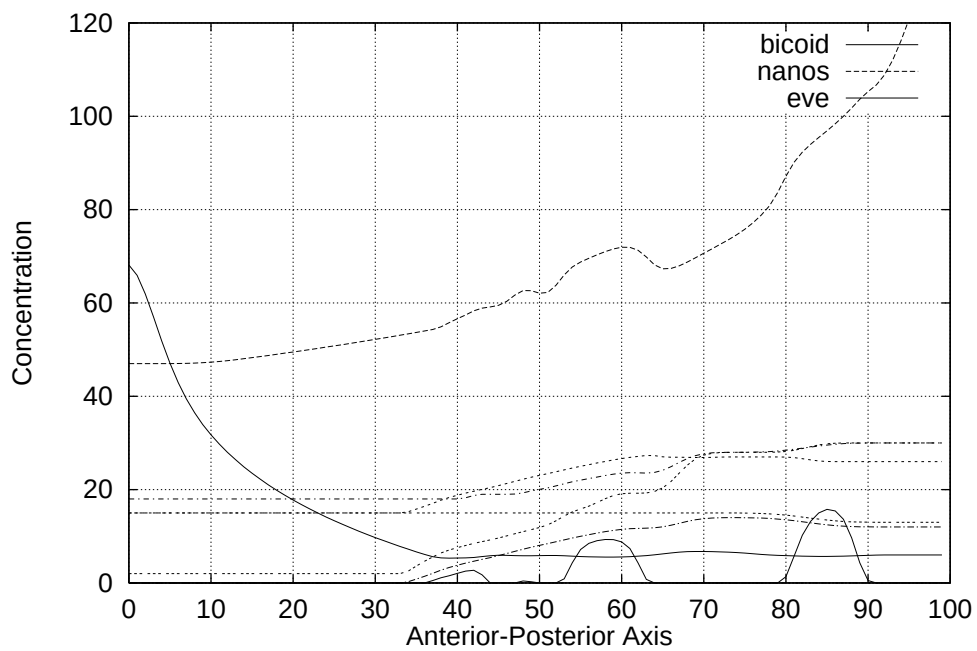


図 B.4: 実験 2 の「拡散-合成-削除」ステップ 15 の各遺伝子調節タンパク質の分布

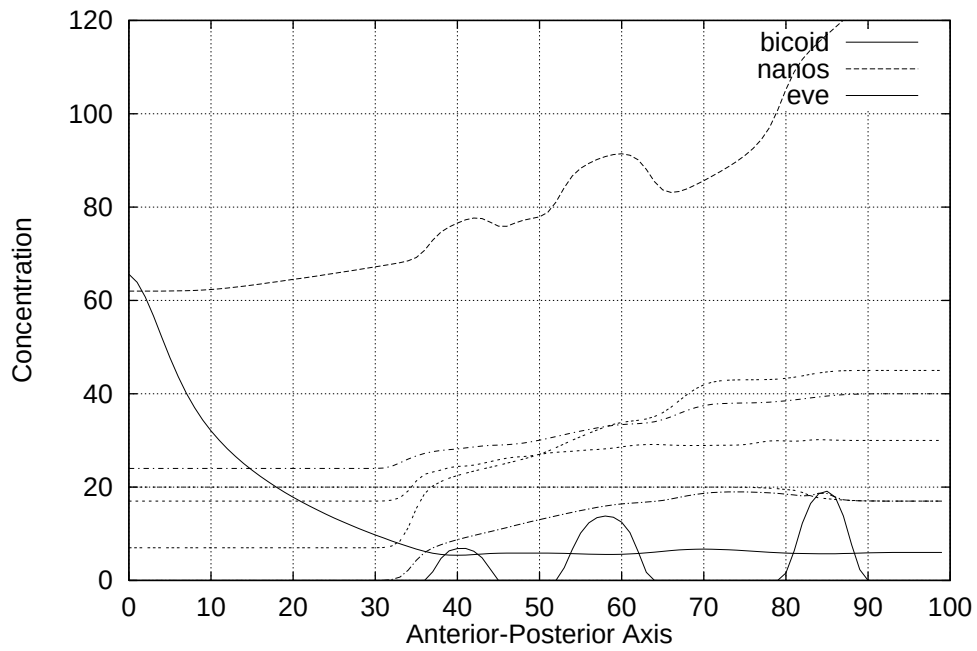


図 B.5: 実験 2 の「拡散-合成-削除」ステップ 20 の各遺伝子調節タンパク質の分布

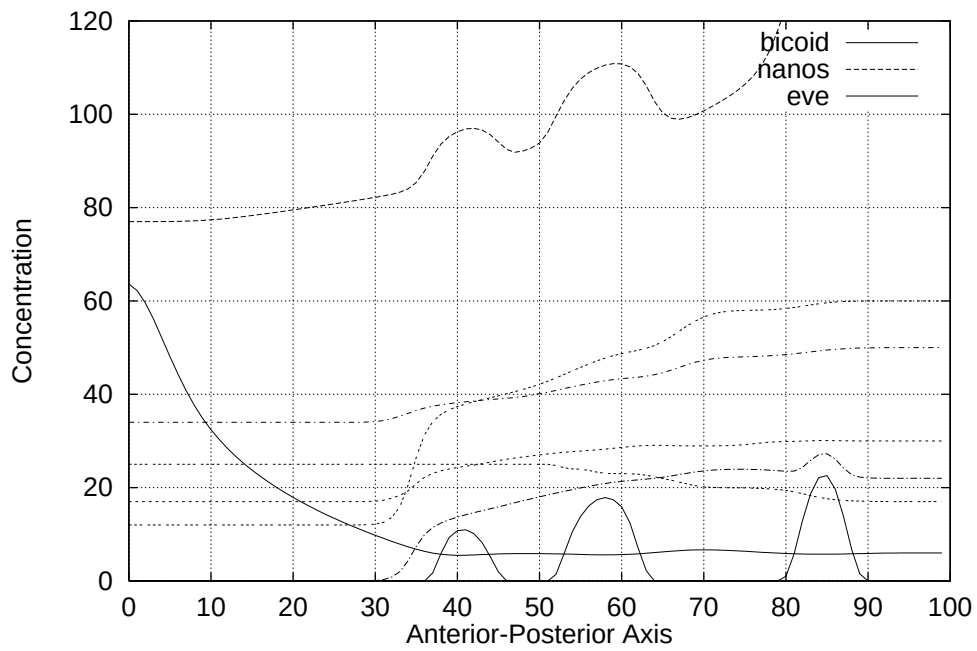


図 B.6: 実験 2 の「拡散-合成-削除」ステップ 25 の各遺伝子調節タンパク質の分布

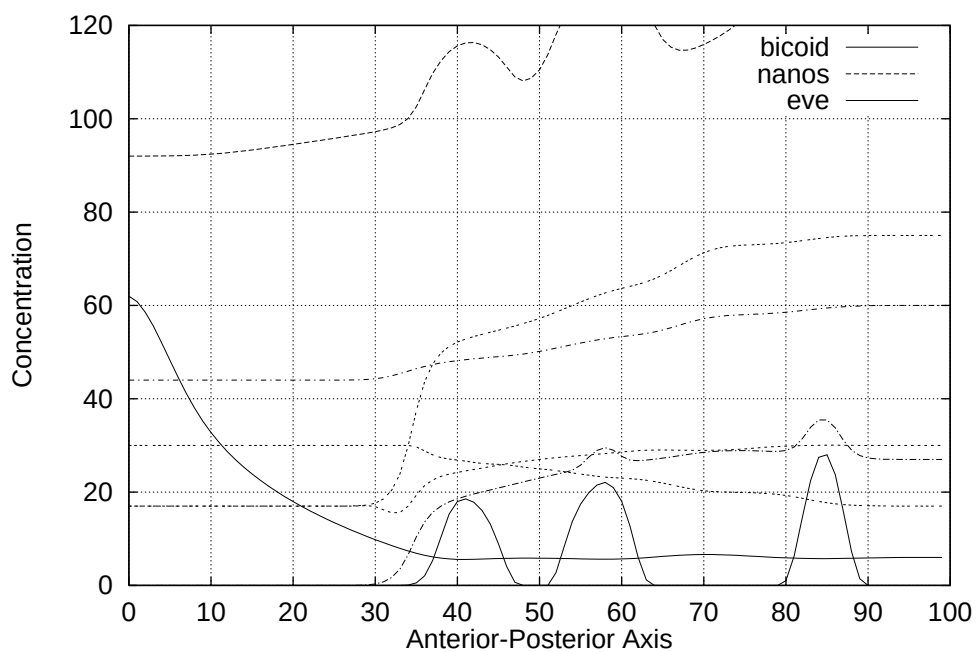


図 B.7: 実験 2 の「拡散-合成-削除」ステップ 30 の各遺伝子調節タンパク質の分布

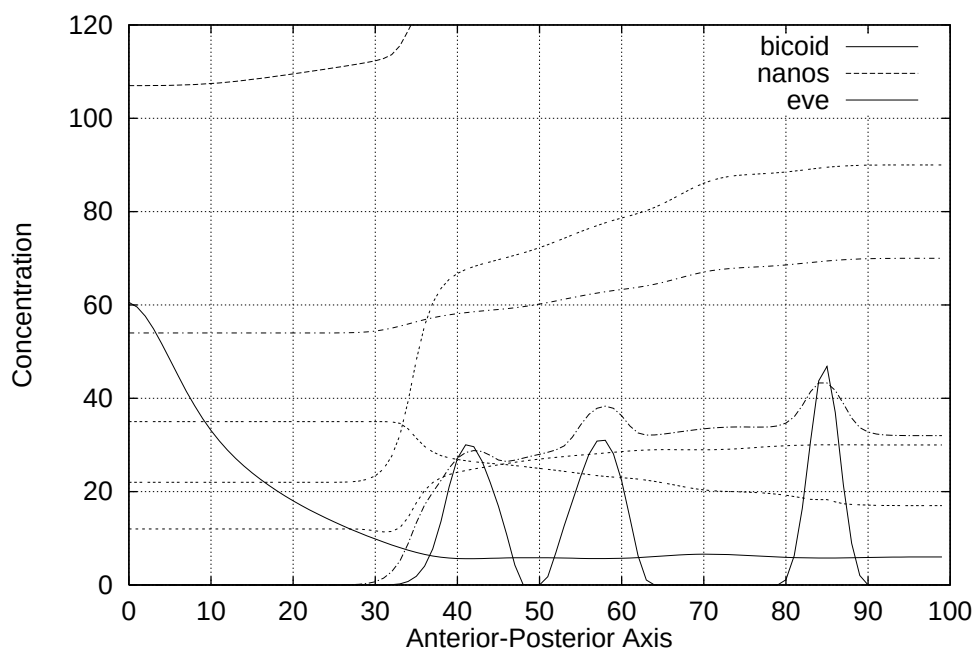


図 B.8: 実験 2 の「拡散-合成-削除」ステップ 35 の各遺伝子調節タンパク質の分布

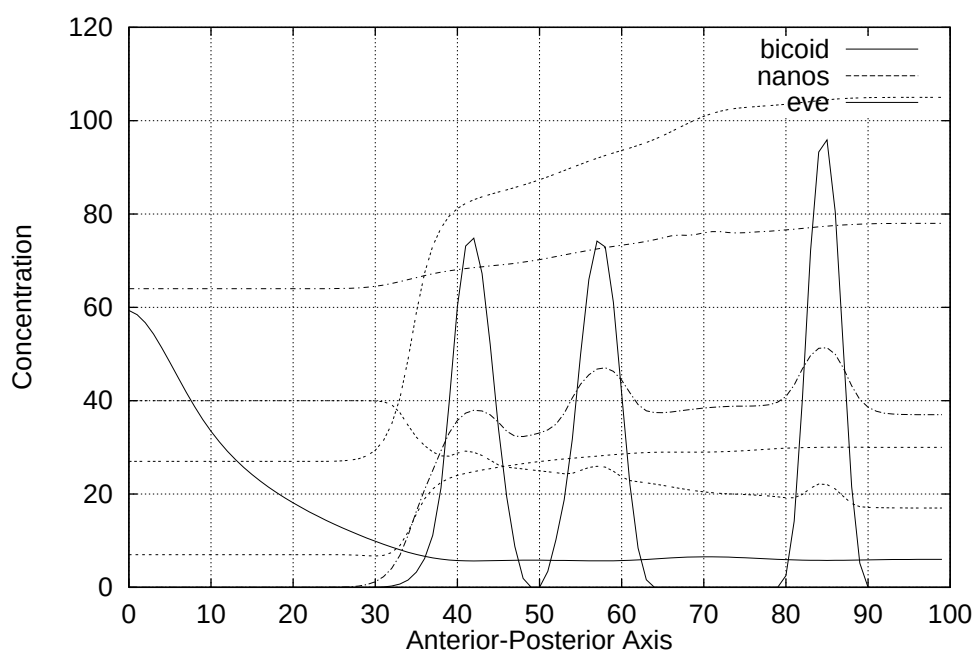


図 B.9: 実験 2 の「拡散-合成-削除」ステップ 40 の各遺伝子調節タンパク質の分布

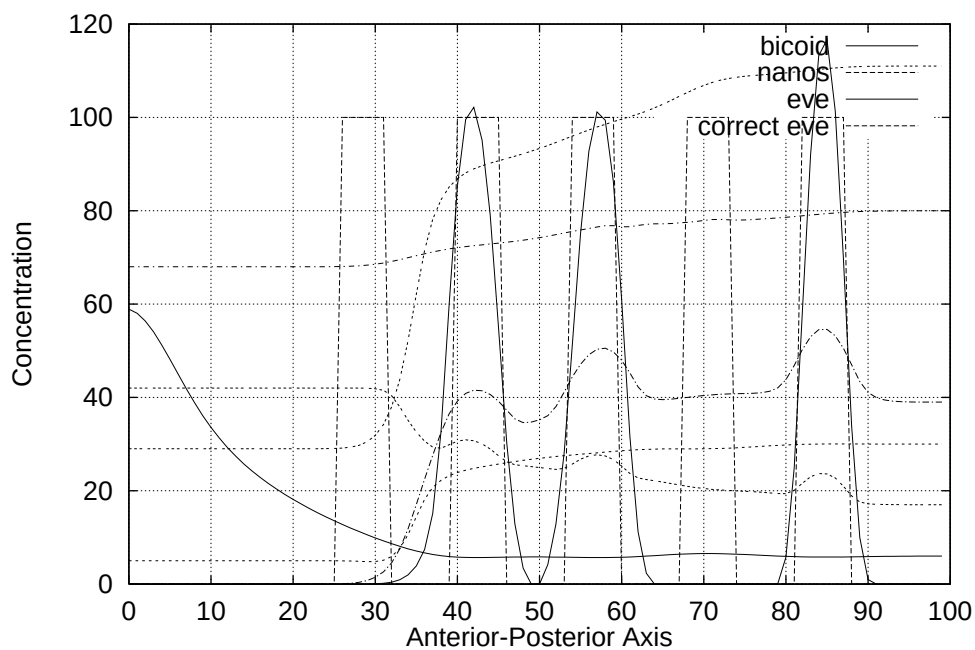


図 B.10: 実験 2 の「拡散-合成-削除」ステップ 42 の各遺伝子調節タンパク質の分布。このステップがもっとも正解と近かったときである。これ以後は正解から離れていく。



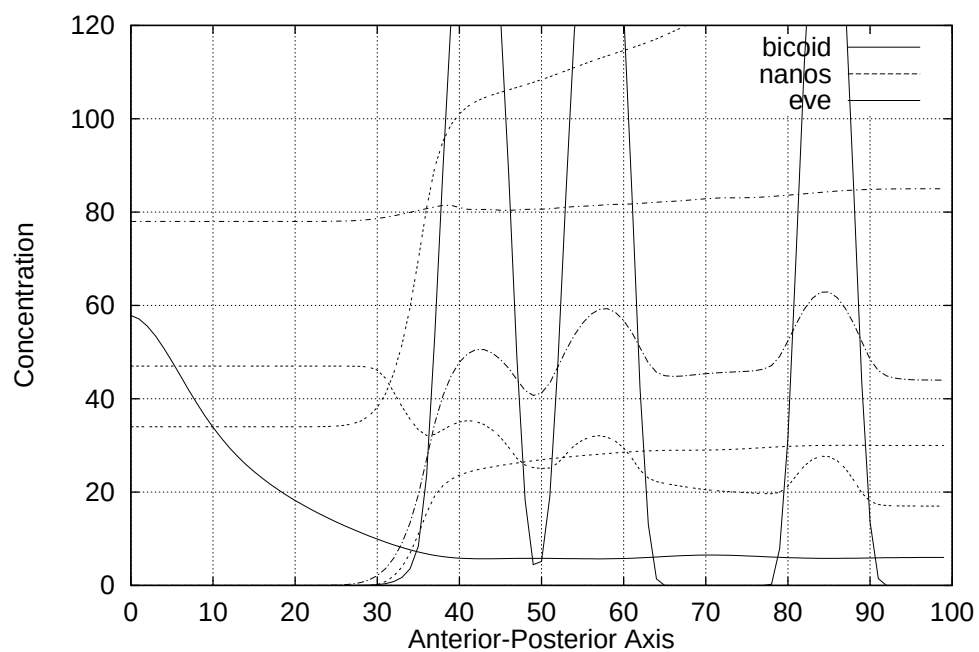


図 B.11: 実験 2 の「拡散-合成-削除」ステップ 47 の各遺伝子調節タンパク質の分布

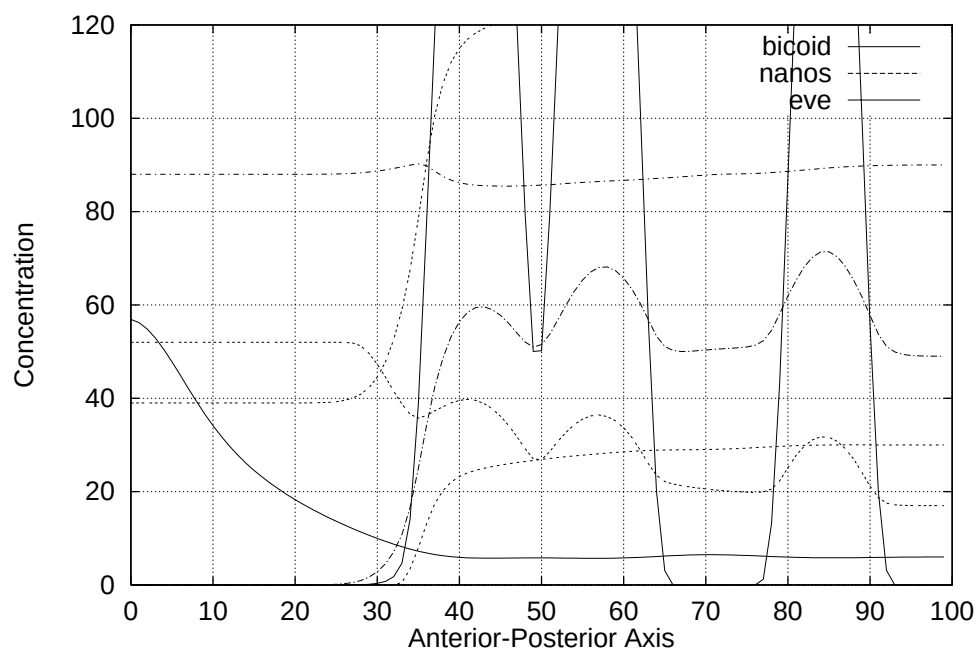


図 B.12: 実験 2 の「拡散-合成-削除」ステップ 52 の各遺伝子調節タンパク質の分布

# 付 録 C

## 実験3

### C.1 実験 3 で得られた最良個体

実験3で得られたの最良個体を以下に示す。なお、“protein0”は bicoid、“protein1”は nanos、“protein9”は eve を示す。

```
(IF (> (ConcentrationOf protein1) 8.829569) protein0)
(IF (< (ConcentrationOf protein0) 10.263046) protein2)
(IF (< (ConcentrationOf protein3) 3.647993) protein7)
(IF (< (ConcentrationOf protein0) 6.717147) protein1)
(IF (< (ConcentrationOf protein0) 16.538759) protein3)
(IF (> (ConcentrationOf protein3) 76.500198) protein2)
(IF (< (ConcentrationOf protein3) 27.621264) protein1)
(IF (< (ConcentrationOf protein3) 3.749130) protein2)
(IF (> (ConcentrationOf protein8) 5.601604) protein1)
(IF (< (ConcentrationOf protein0) 6.717147) protein5)
(IF (< (ConcentrationOf protein0) 53.012764) protein7)
(IF (< (ConcentrationOf protein0) 36.508282) protein3)
(IF (< (ConcentrationOf protein5) 31.500433) protein0)
(IF (> (ConcentrationOf protein8) 3.749130) protein1)
(IF (> (ConcentrationOf protein8) 71.507103) protein1)
(IF (< (ConcentrationOf protein6) 33.016197) protein3)
```

```

(IF (> (ConcentrationOf protein4) 16.909220) protein6)
(IF (< (ConcentrationOf protein3) 5.287442) protein3)
(IF (< (ConcentrationOf protein6) 48.541447) protein6)
(IF (> (ConcentrationOf protein6) 35.450871) protein1)
(IF (< (ConcentrationOf protein7) 53.012764) protein1)
(IF (< (ConcentrationOf protein5) 0.082797) protein7)
(IF (> (ConcentrationOf protein2) 72.907440) protein8)
(IF (> (ConcentrationOf protein3) 0.823150) protein5)
(IF (< (ConcentrationOf protein3) 11.062551) protein6)
(IF (< (ConcentrationOf protein0) 6.717147) protein5)
(IF (> (ConcentrationOf protein6) 84.924538) protein5)
(IF (< (ConcentrationOf protein0) 6.717147) protein5)
(IF (> (ConcentrationOf protein5) 50.717918) protein3)
(IF (> (ConcentrationOf protein8) 71.507103) protein1)
(IF (< (ConcentrationOf protein3) 4.938119) protein5)
(IF (> (ConcentrationOf protein8) 5.601604) protein1)
(IF (< (ConcentrationOf protein0) 10.779678) protein3)
(IF (> (ConcentrationOf protein5) 47.933983) protein6)
(IF (< (ConcentrationOf protein0) 10.779678) protein3)
(IF (> (ConcentrationOf protein8) 2.840865) protein7)
(IF (< (ConcentrationOf protein1) 42.815163) protein7)
(IF (> (ConcentrationOf protein2) 22.060305) protein4)

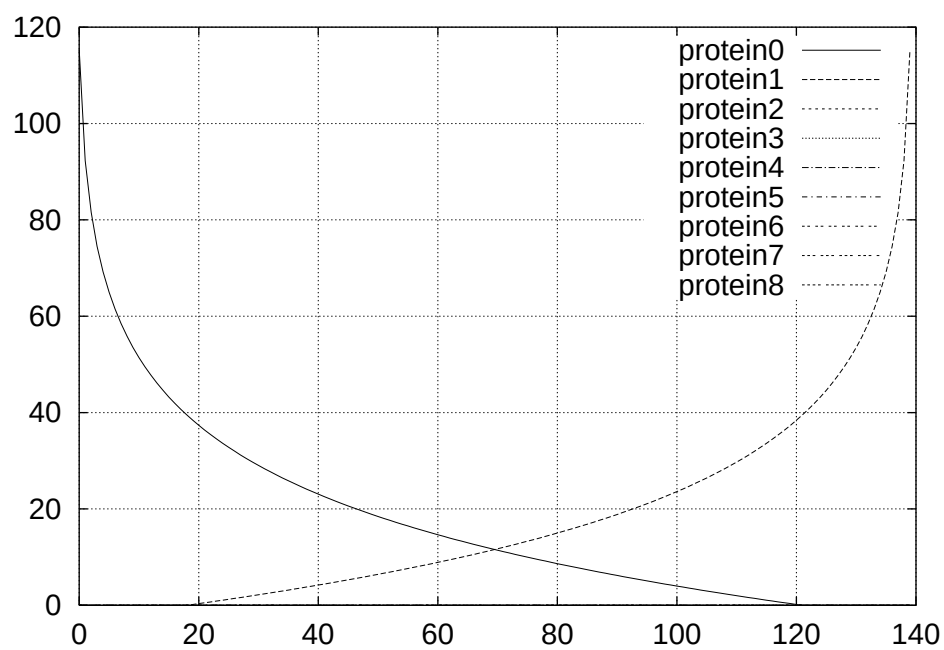
```

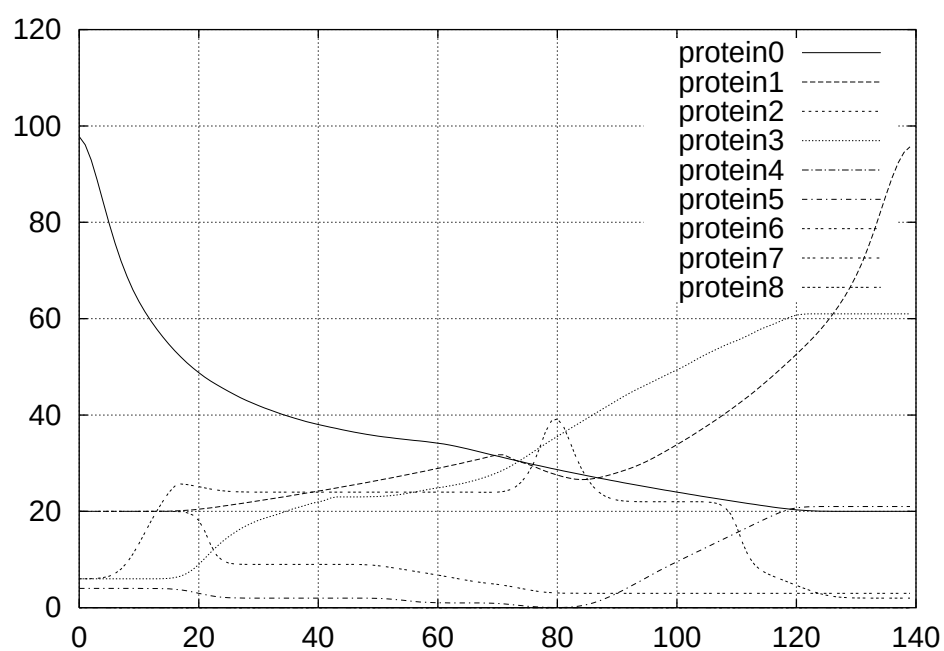
## C.2 最良個体の遺伝子調節タンパク質の分布の推移

実験3で得られた最良個体は、付録 C.1 に示した通りである。この個体を評価してもっとも正解に近づいたステップは 489 ステップ目であった。その時の各遺伝子調節タンパク質の分布を図 C.2 に示す。

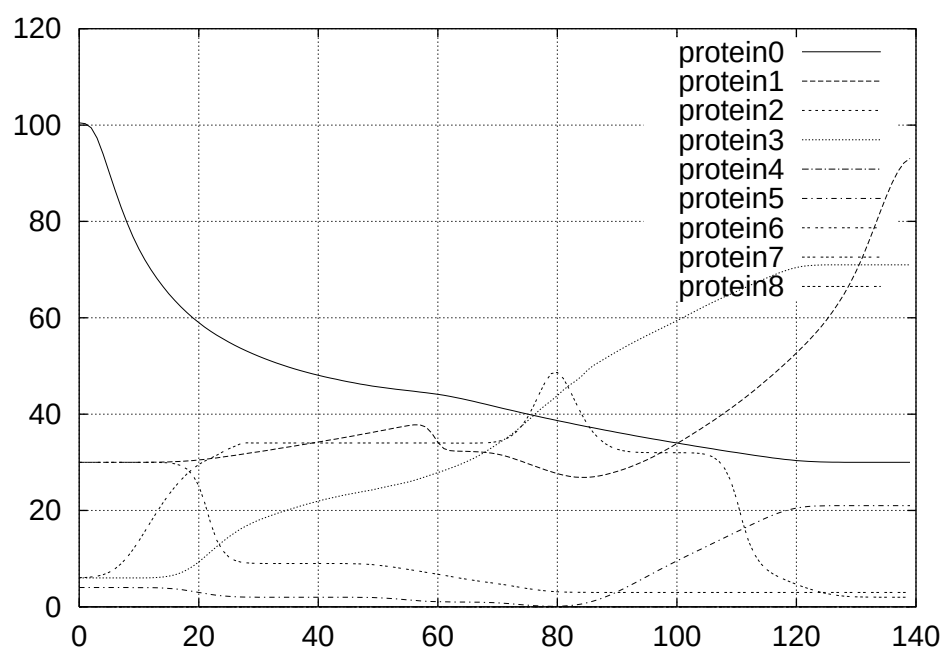
以下では、「合成-拡散-削除」ステップごとの遺伝子調節タンパク質の分布の推移を一

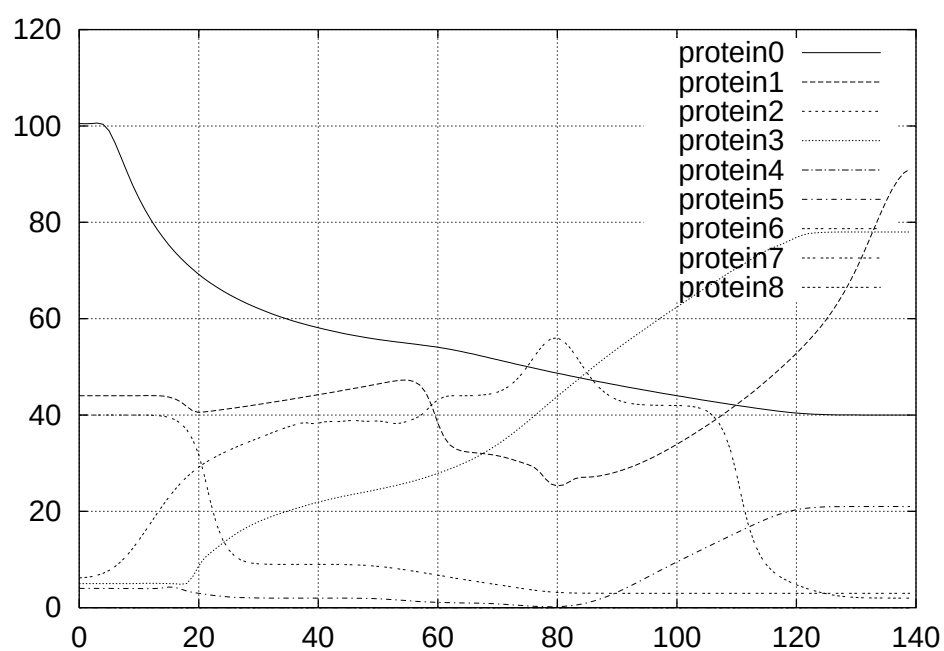
連の図を挙げて示す。



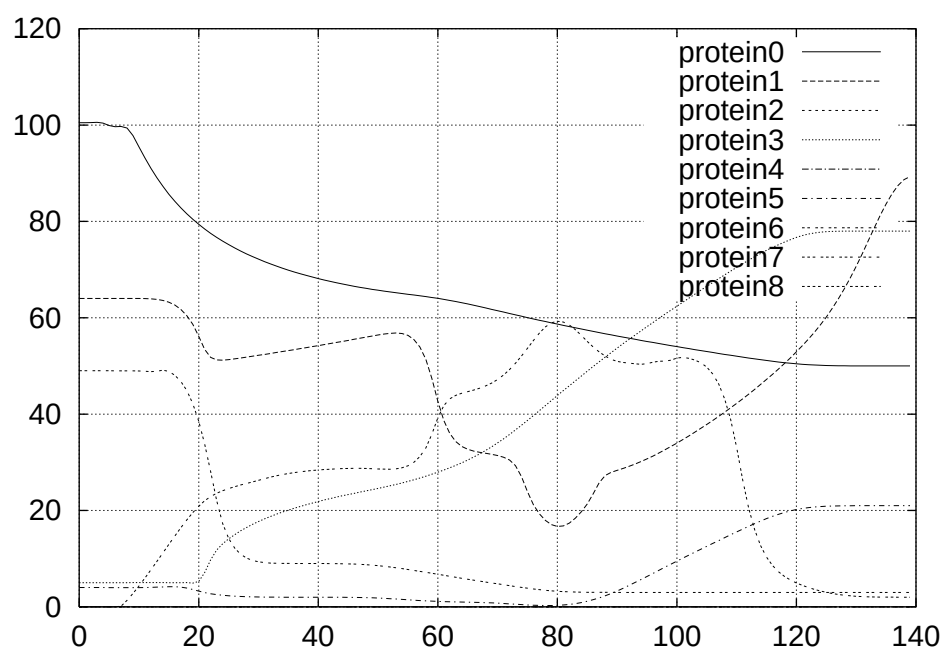


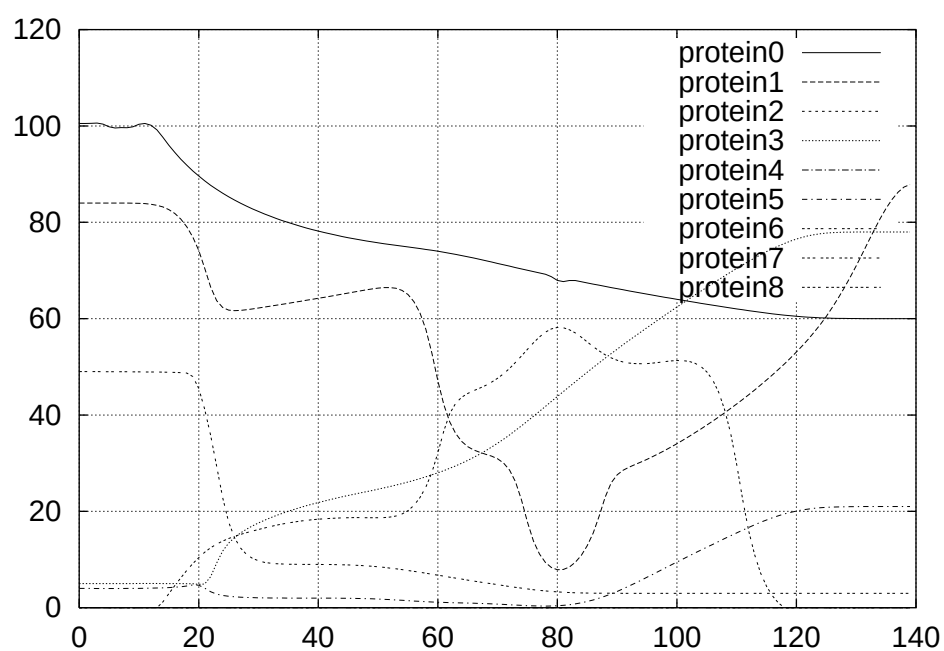
実験3の「拡散-合成-削除」ステップ20の各遺伝子調節タンパク質の分布



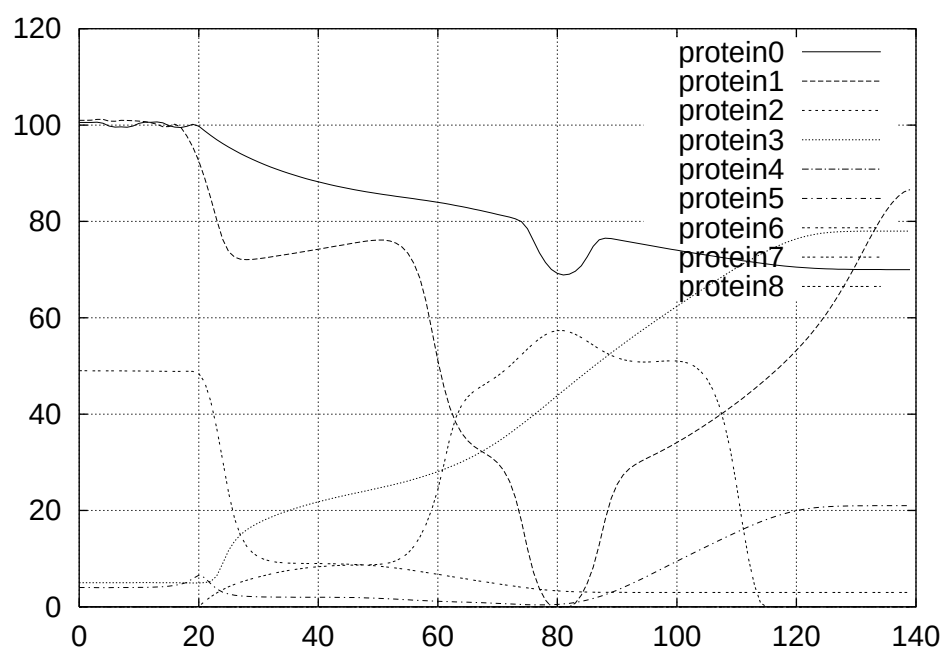


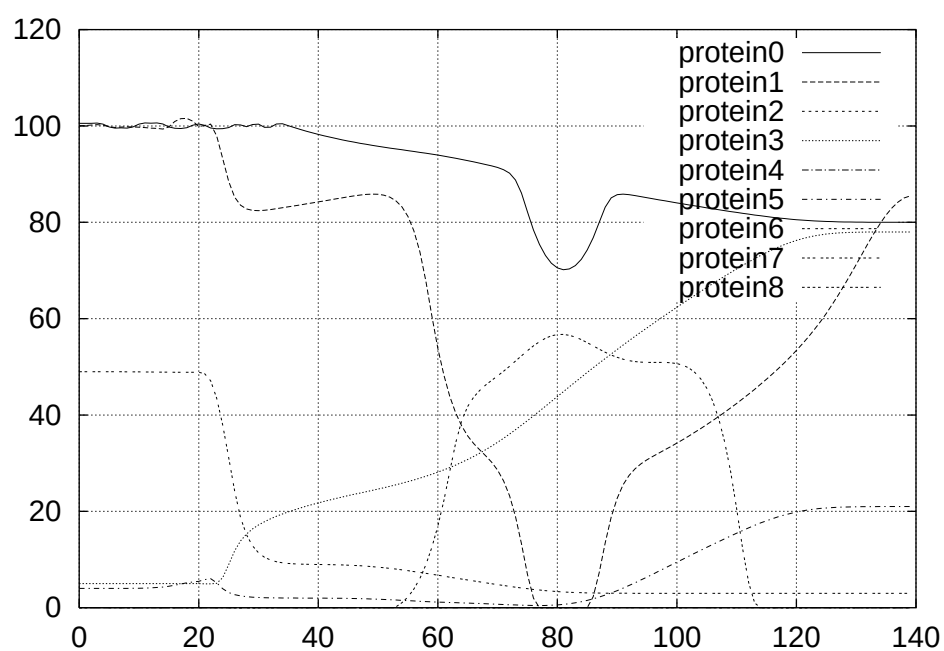
実験3の「拡散-合成-削除」ステップ40の各遺伝子調節タンパク質の分布



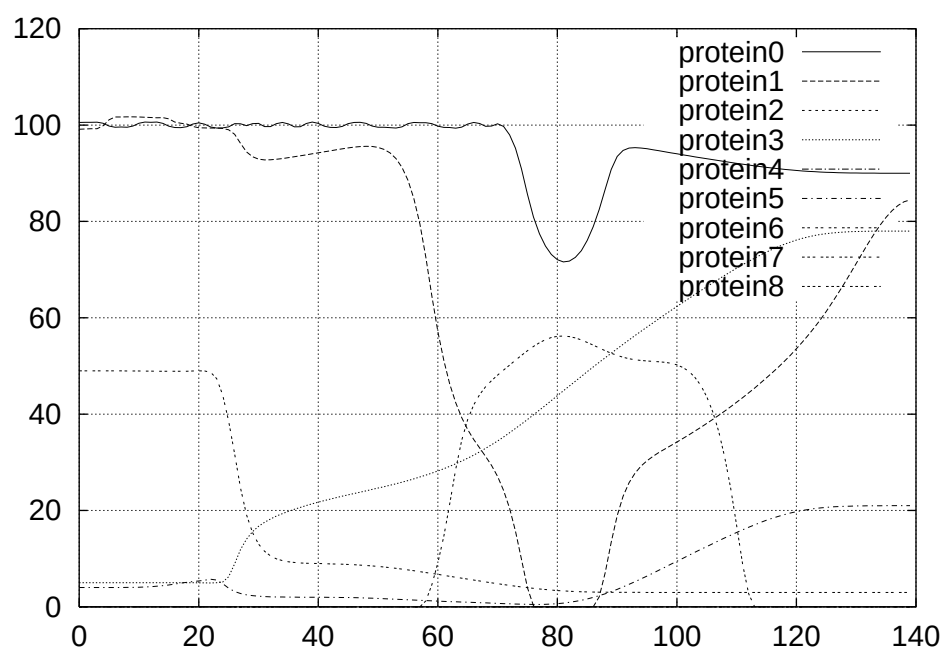


実験3の「拡散-合成-削除」ステップ60の各遺伝子調節タンパク質の分布

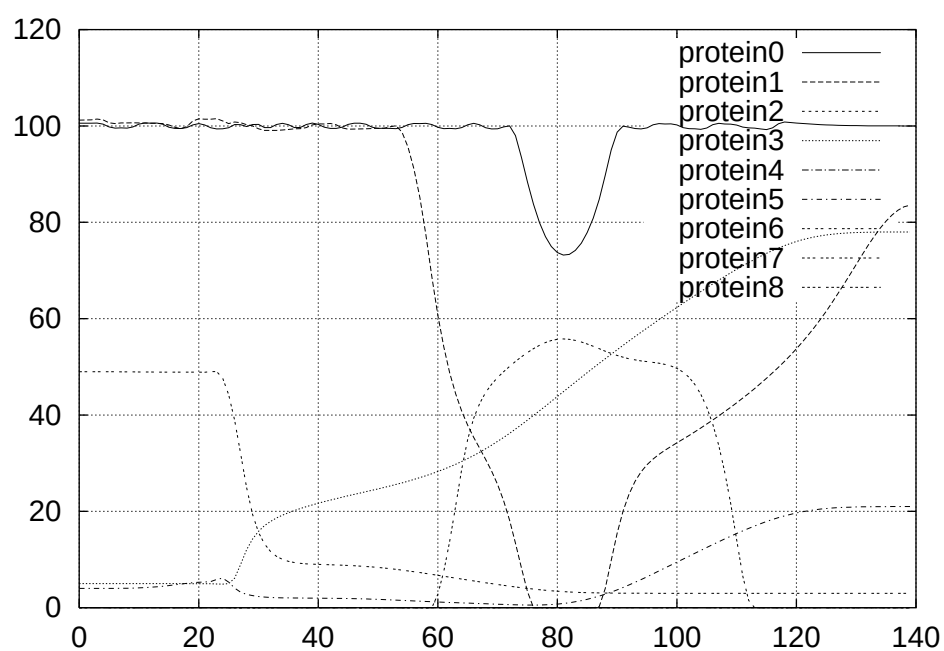




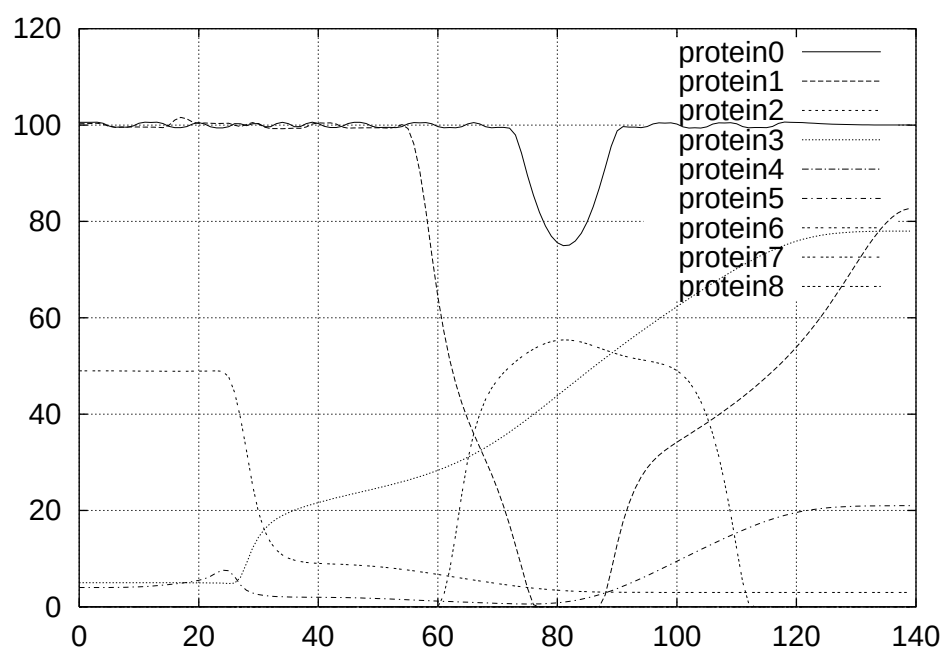
実験3の「拡散-合成-削除」ステップ80の各遺伝子調節タンパク質の分布

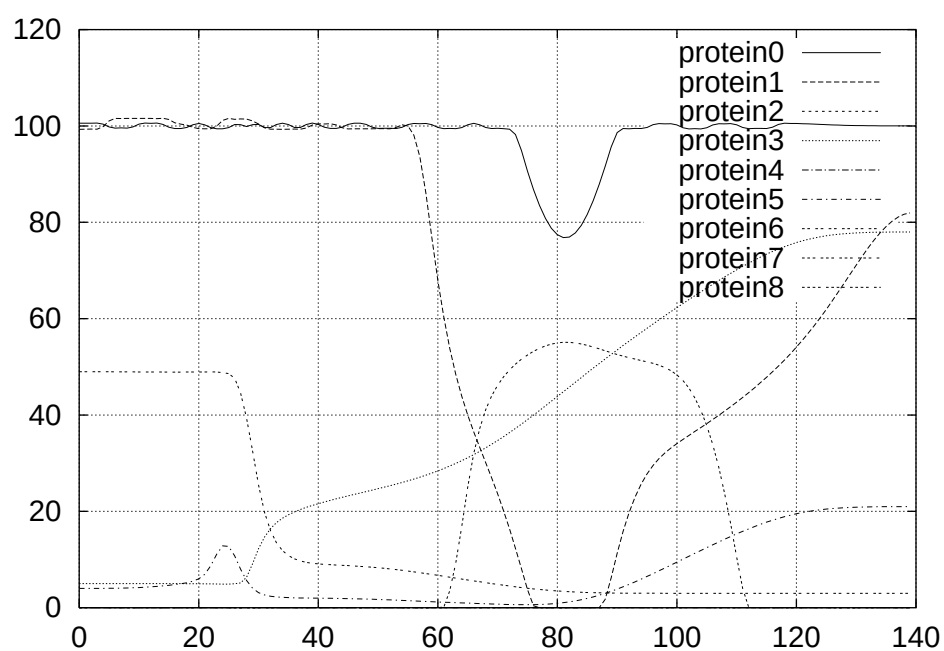




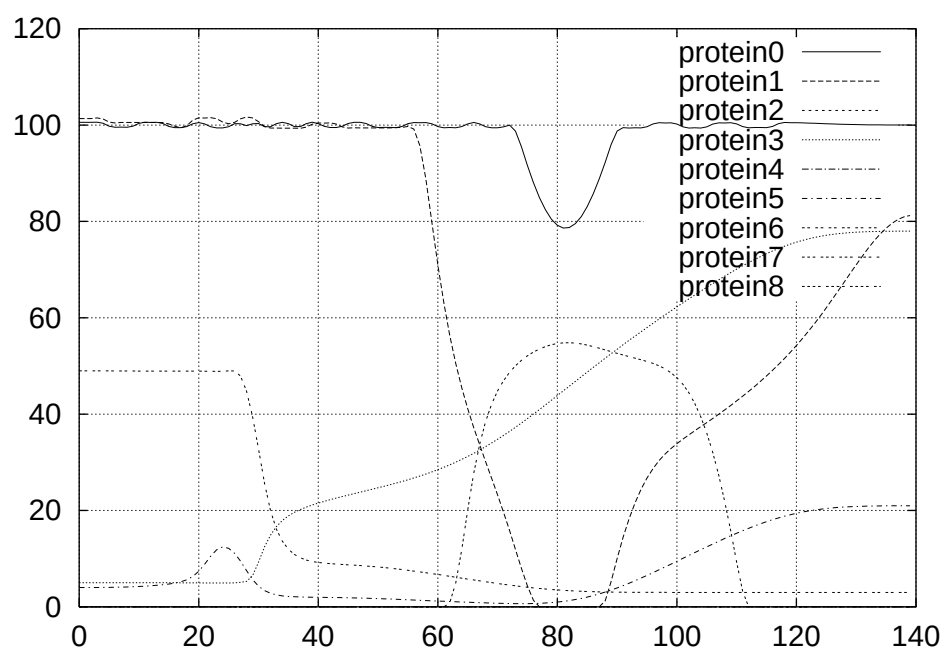


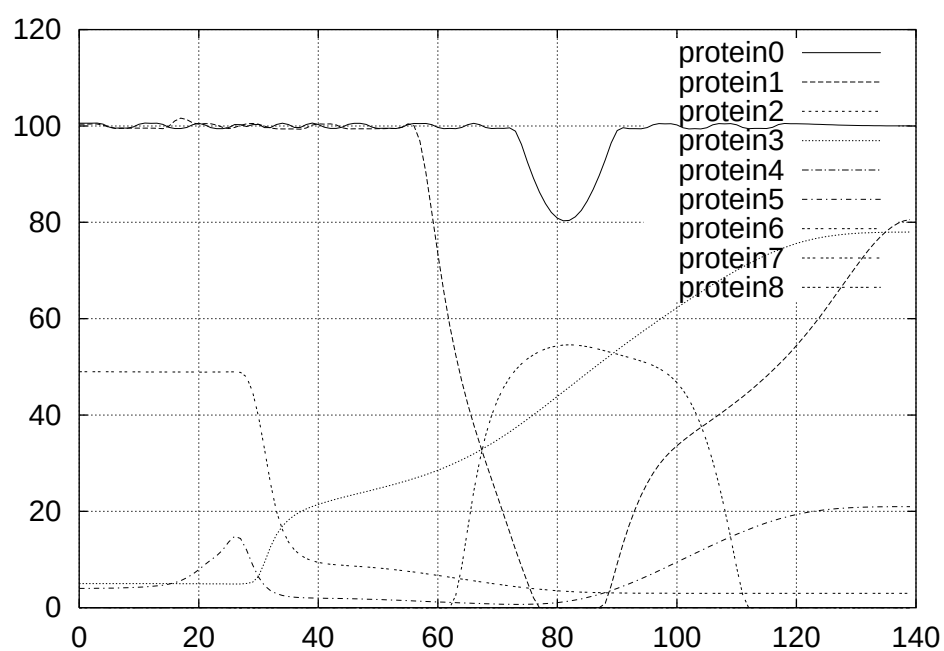
実験3の「拡散-合成-削除」ステップ100の各遺伝子調節タンパク質の分布



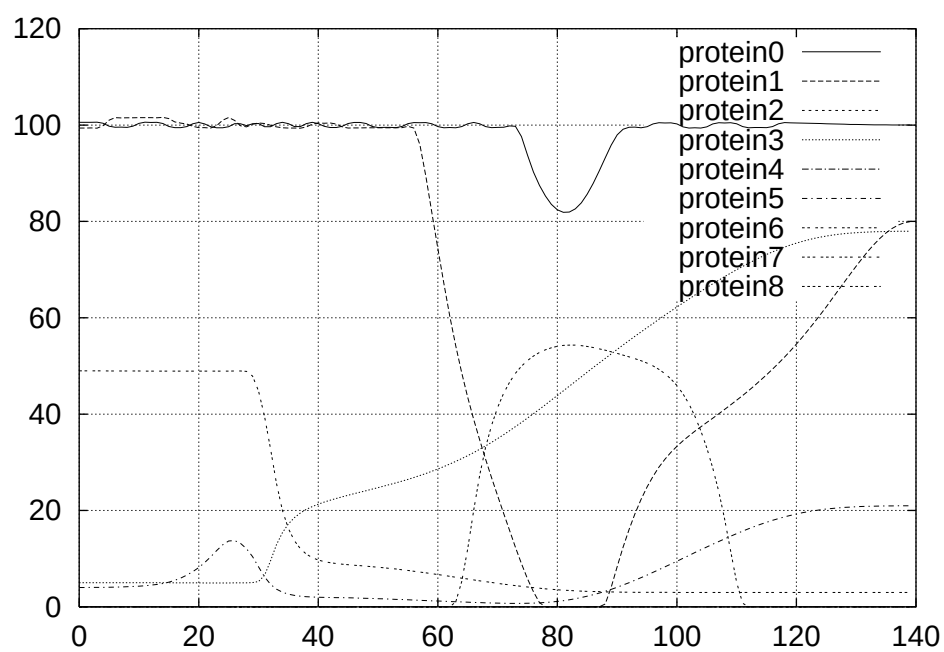


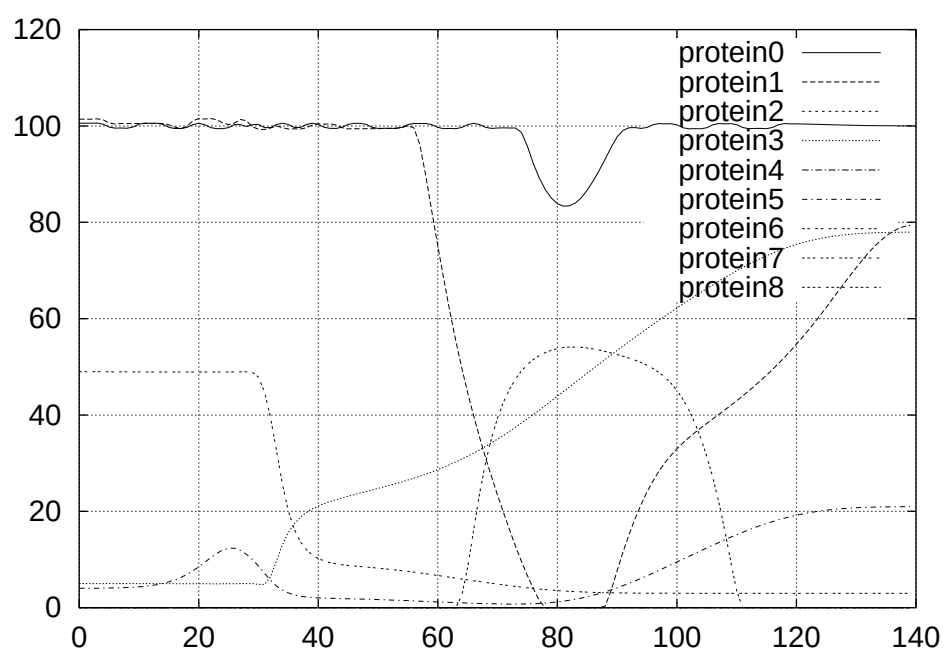
実験3の「拡散-合成-削除」ステップ120の各遺伝子調節タンパク質の分布



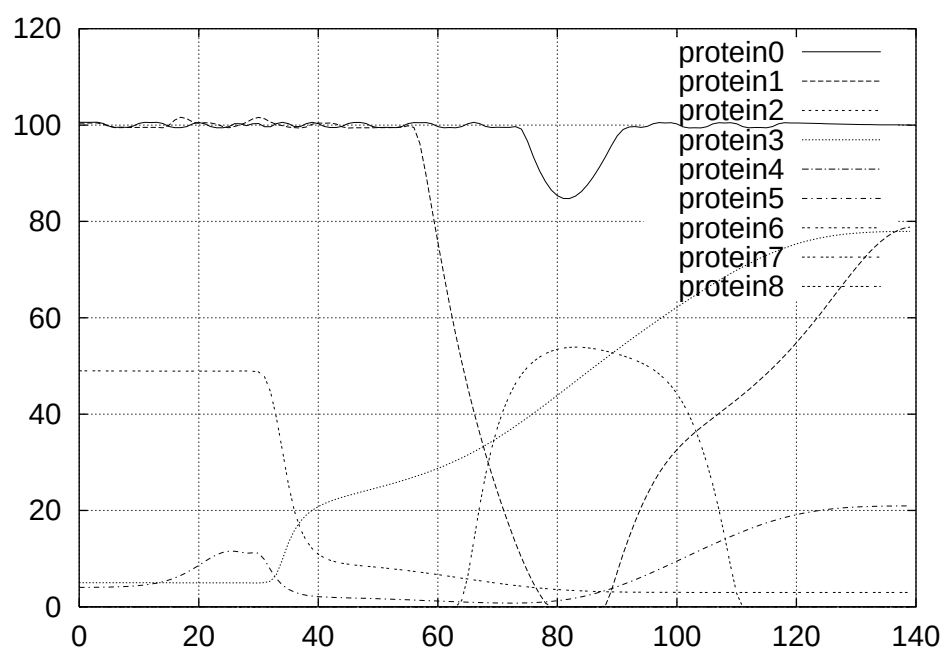


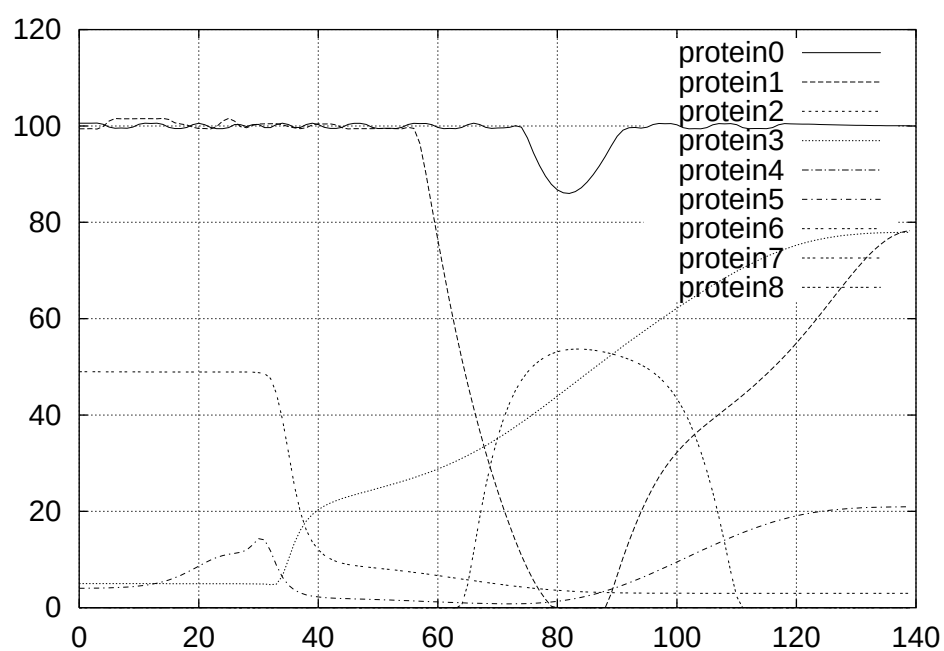
実験3の「拡散-合成-削除」ステップ140の各遺伝子調節タンパク質の分布



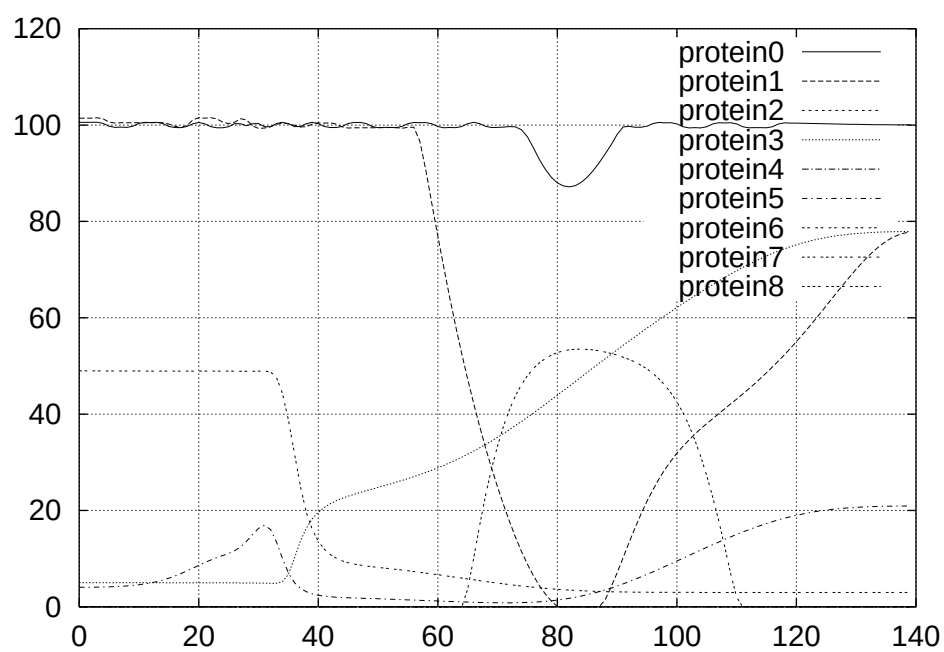


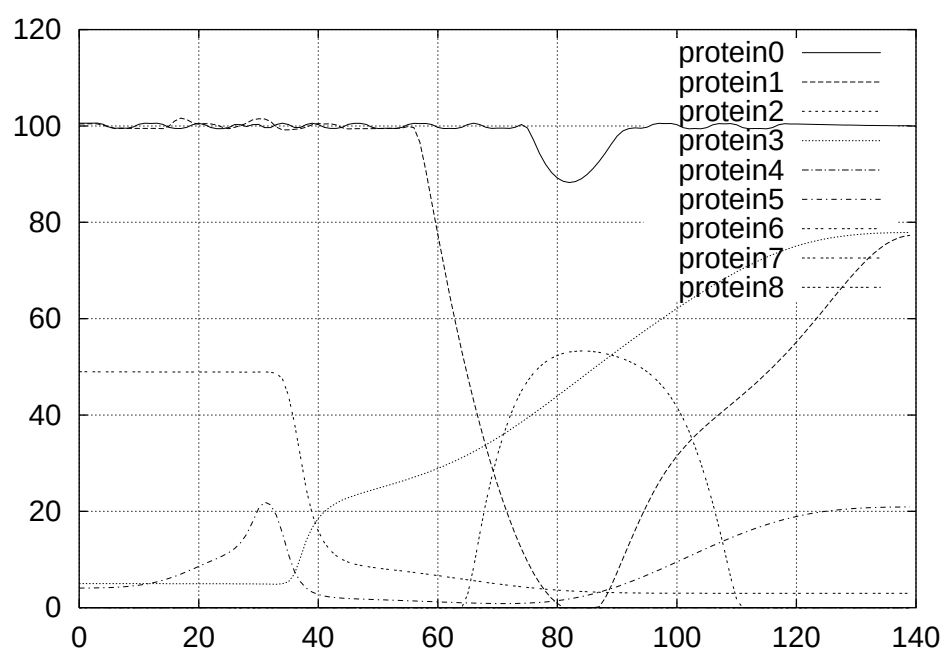
実験3の「拡散-合成-削除」ステップ160の各遺伝子調節タンパク質の分布



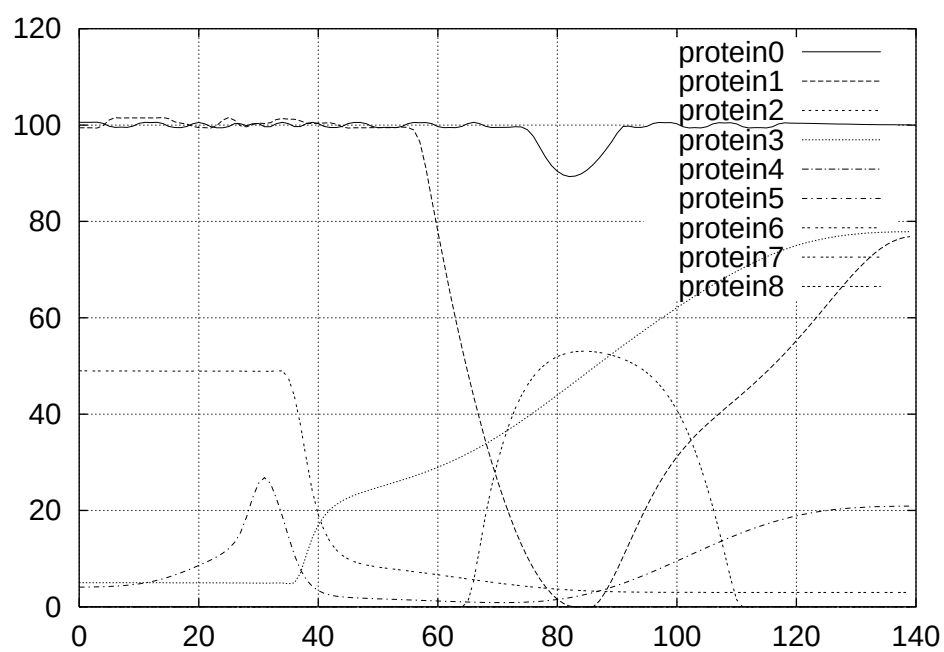


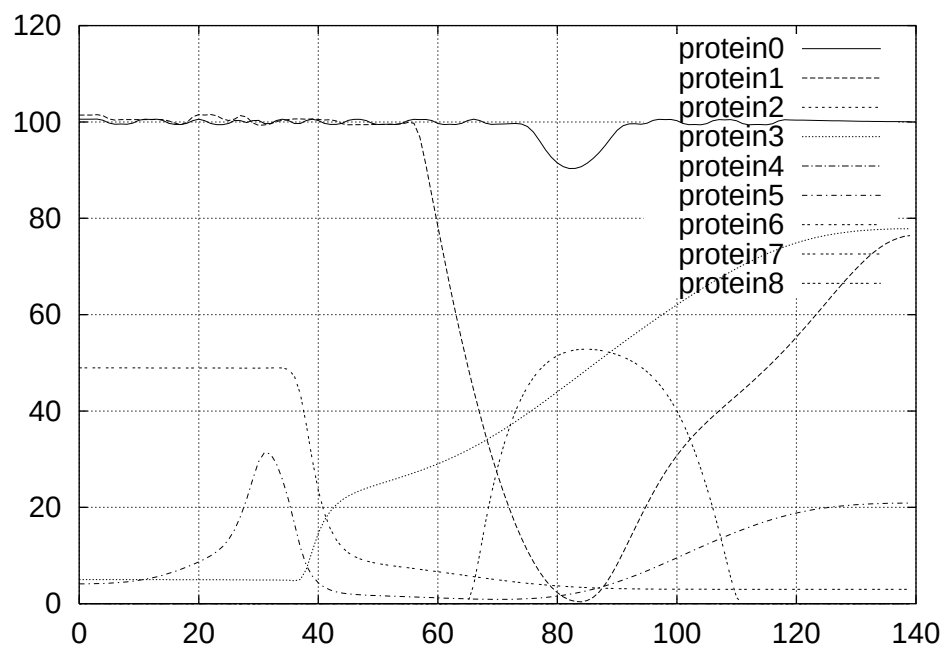
実験3の「拡散-合成-削除」ステップ180の各遺伝子調節タンパク質の分布



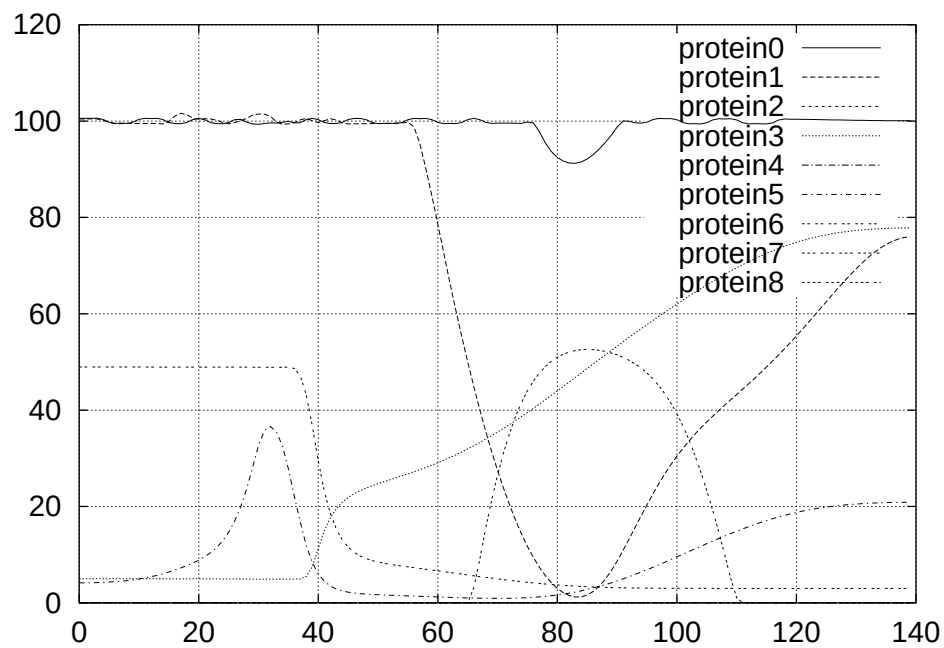


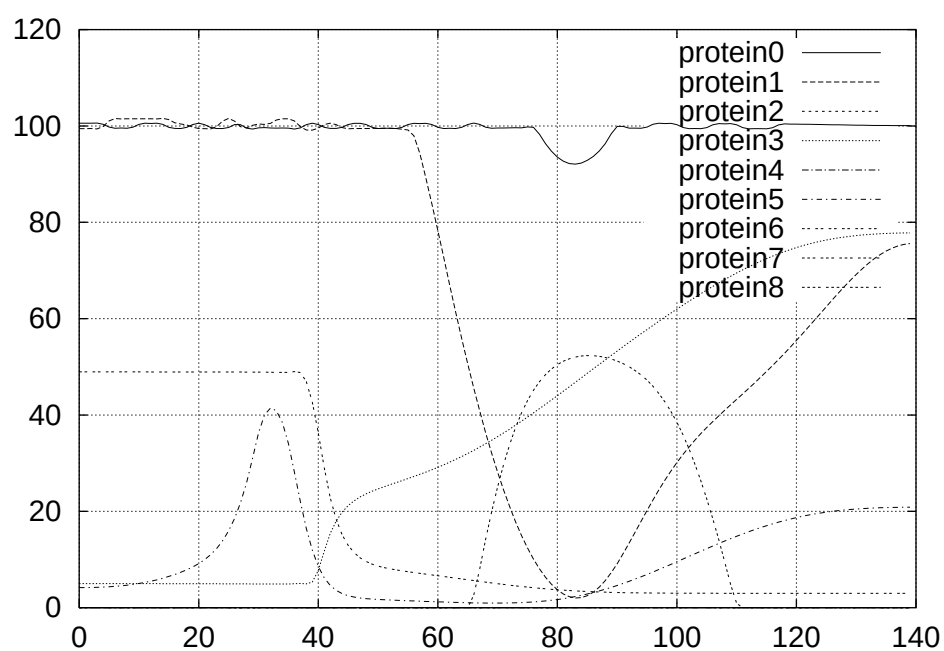
実験3の「拡散-合成-削除」ステップ200の各遺伝子調節タンパク質の分布



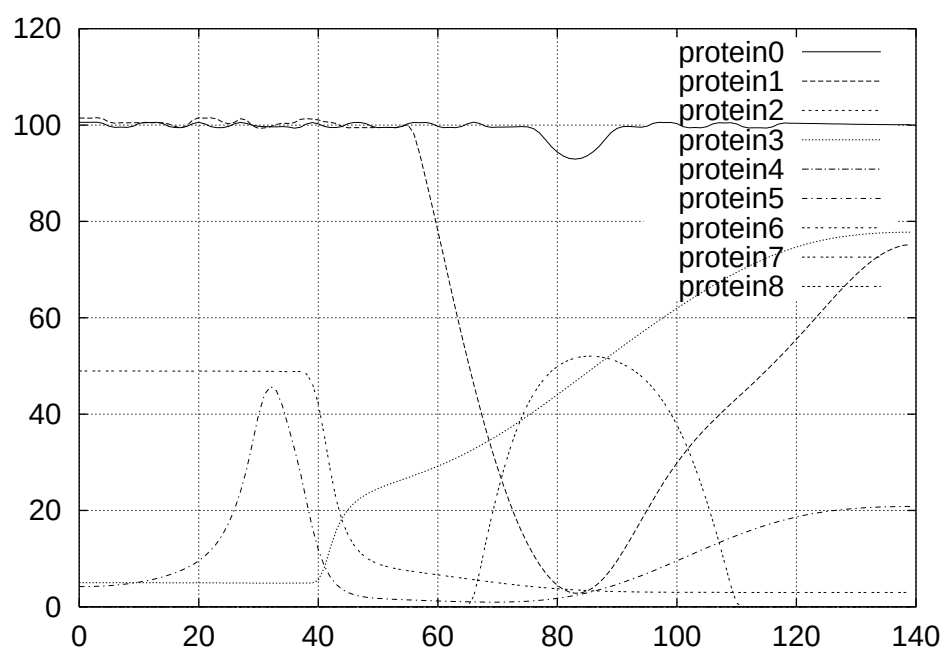


実験3の「拡散-合成-削除」ステップ220の各遺伝子調節タンパク質の分布

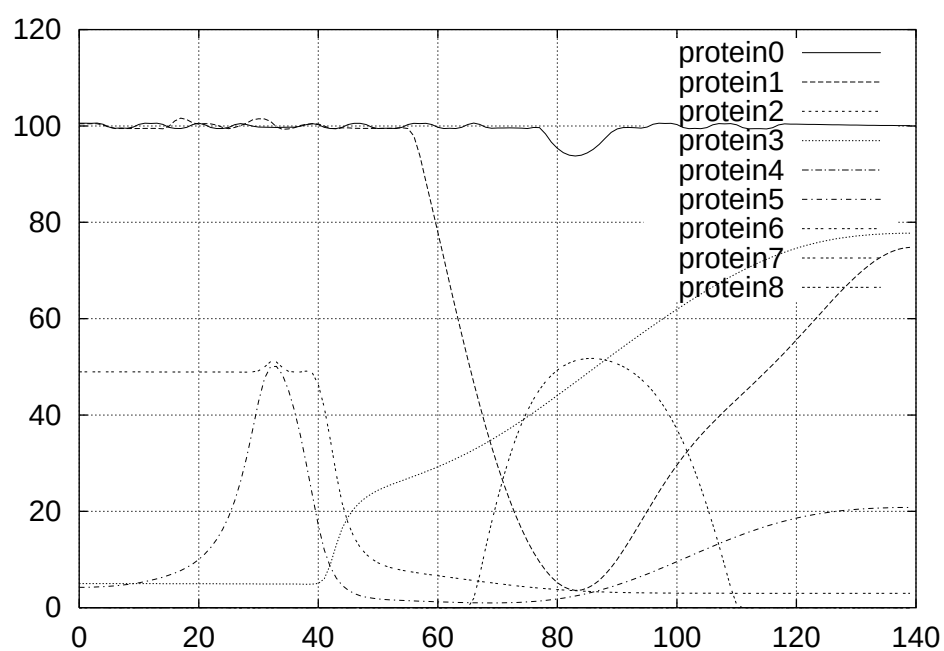




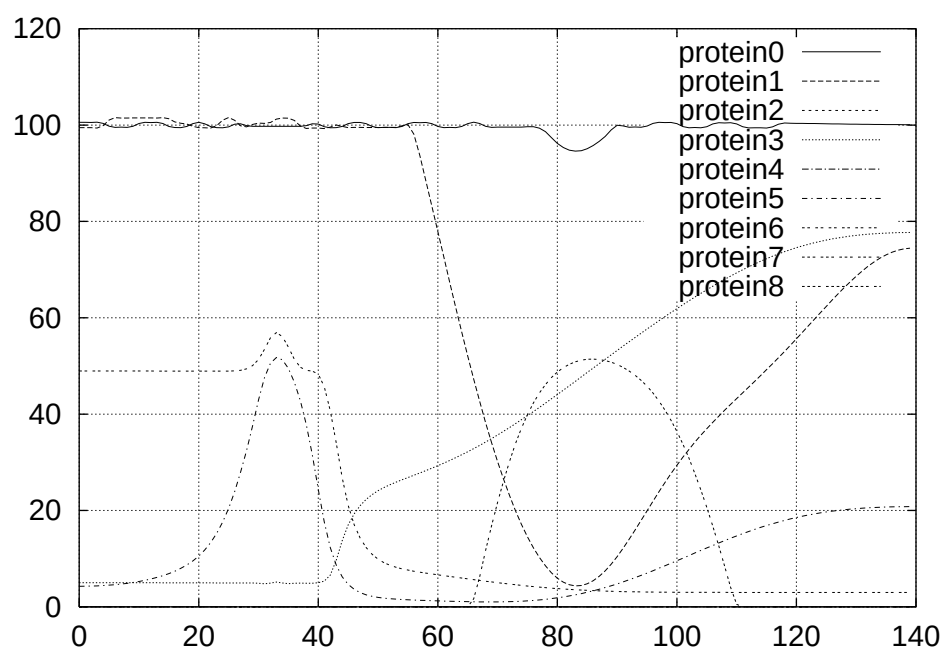
実験3の「拡散-合成-削除」ステップ240の各遺伝子調節タンパク質の分布

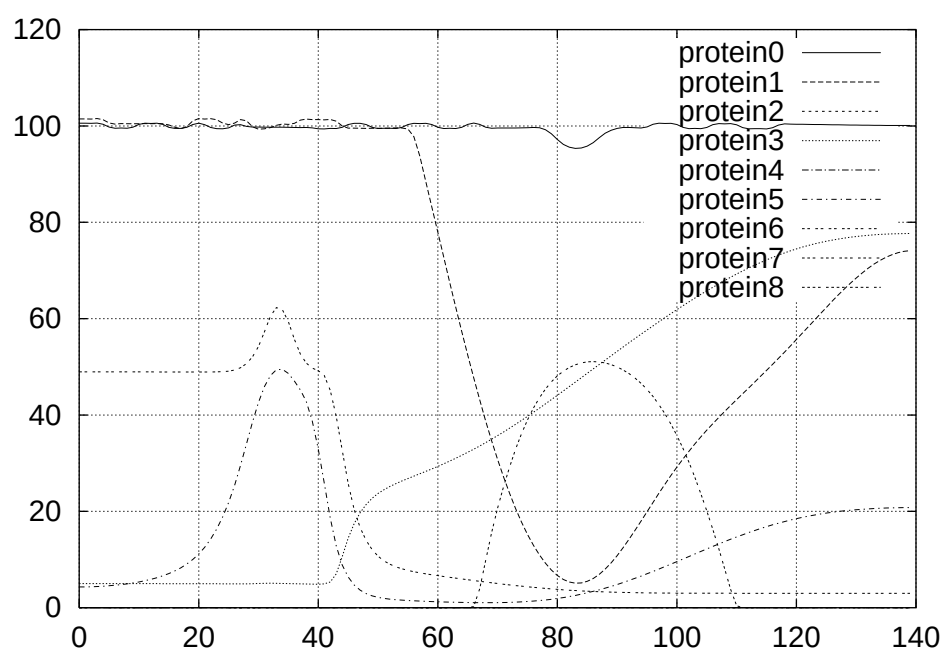




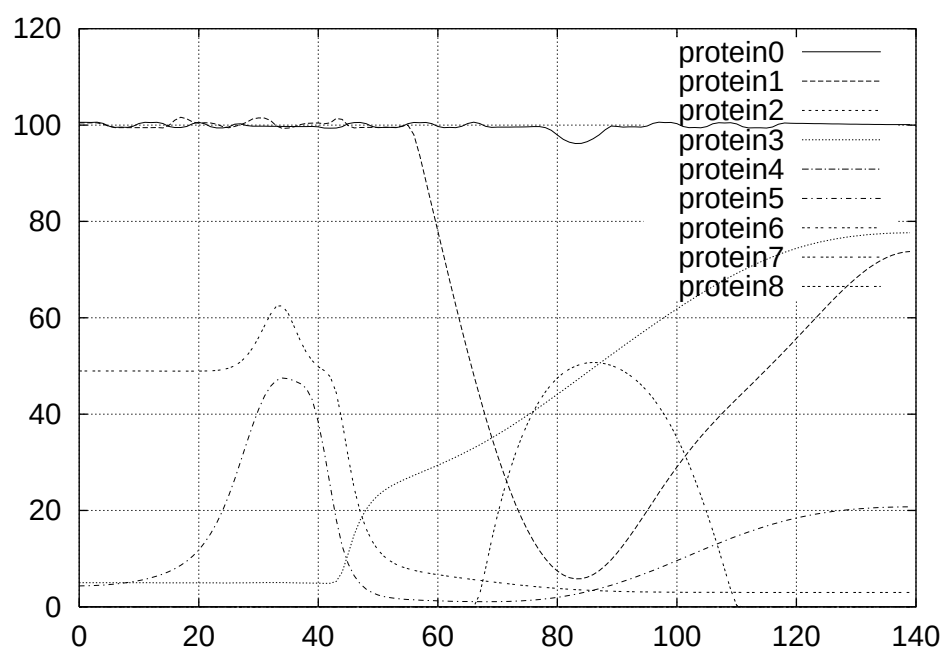


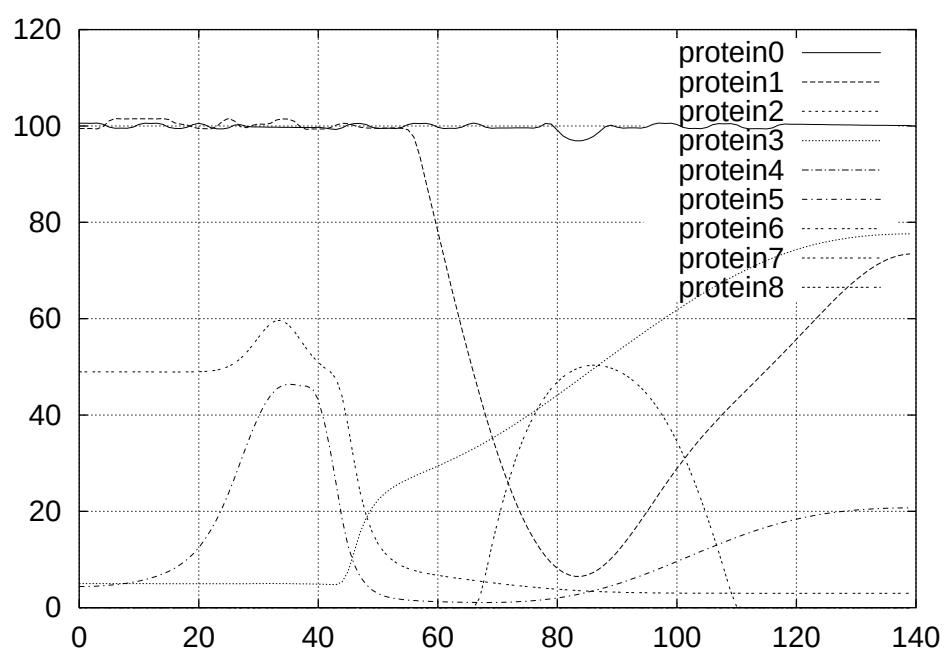
実験3の「拡散-合成-削除」ステップ260の各遺伝子調節タンパク質の分布



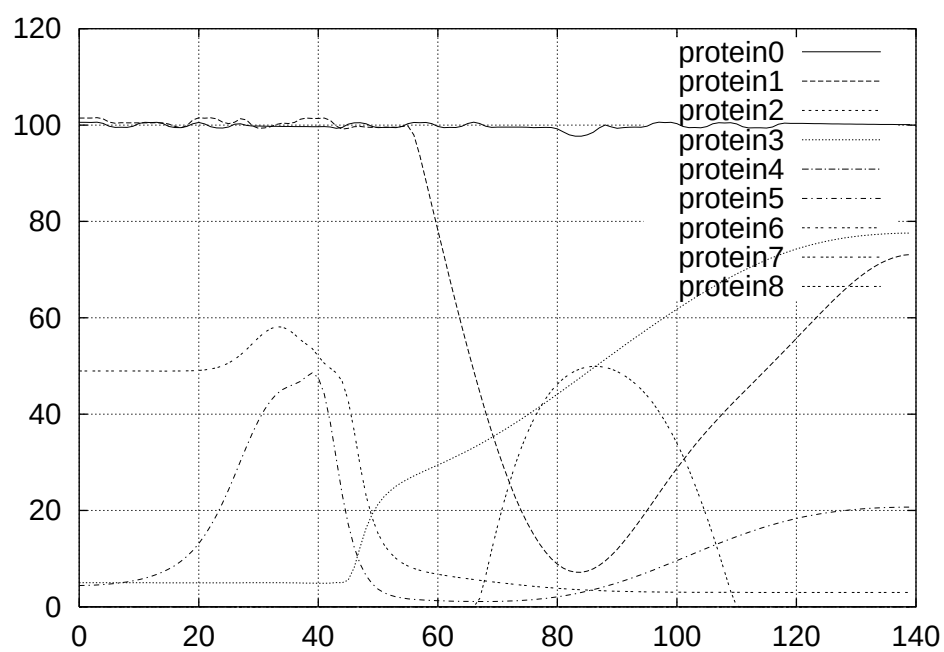


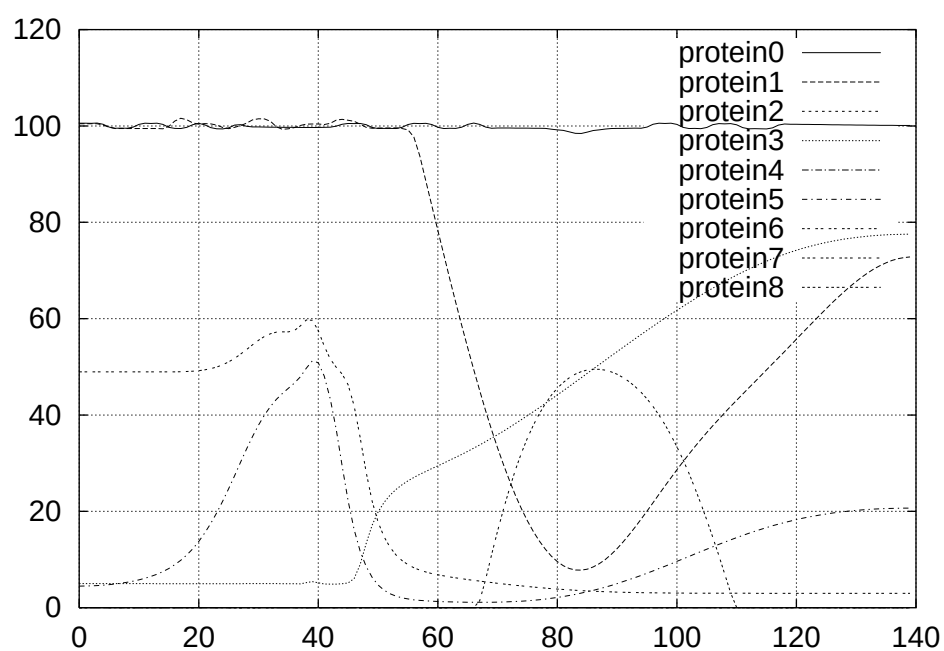
実験3の「拡散-合成-削除」ステップ280の各遺伝子調節タンパク質の分布



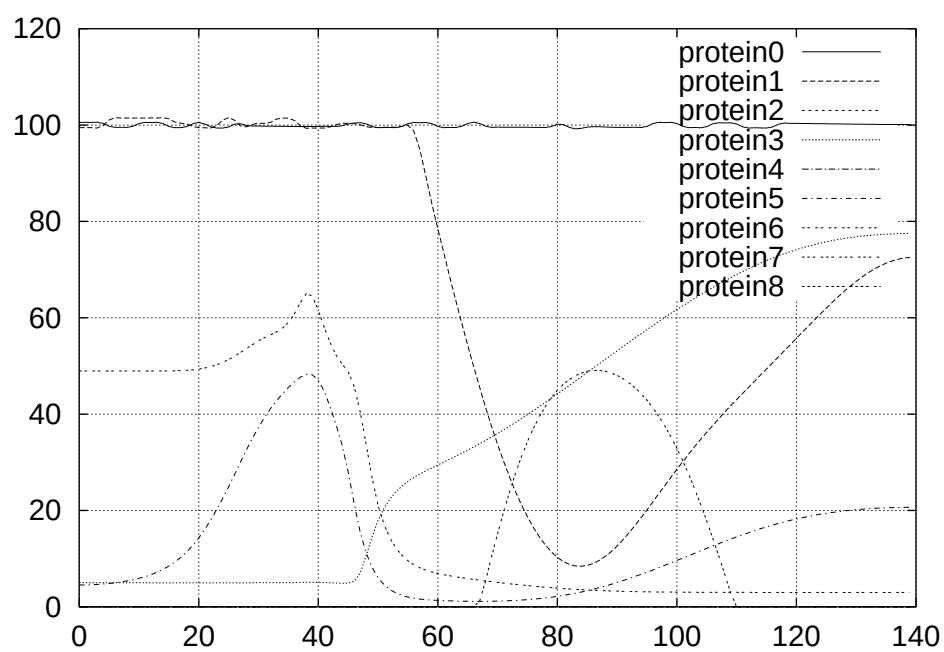


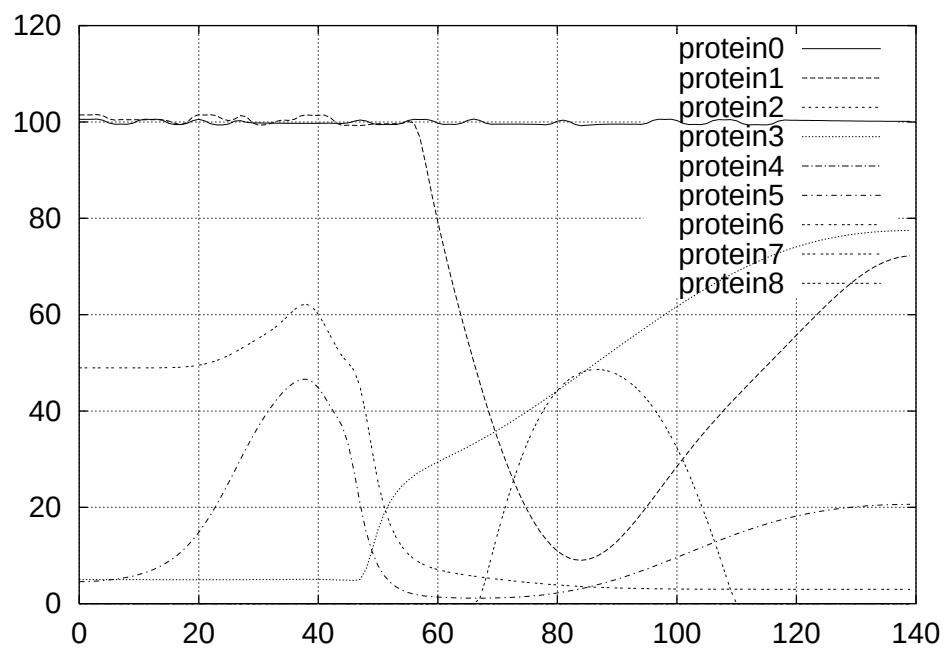
実験3の「拡散-合成-削除」ステップ300の各遺伝子調節タンパク質の分布



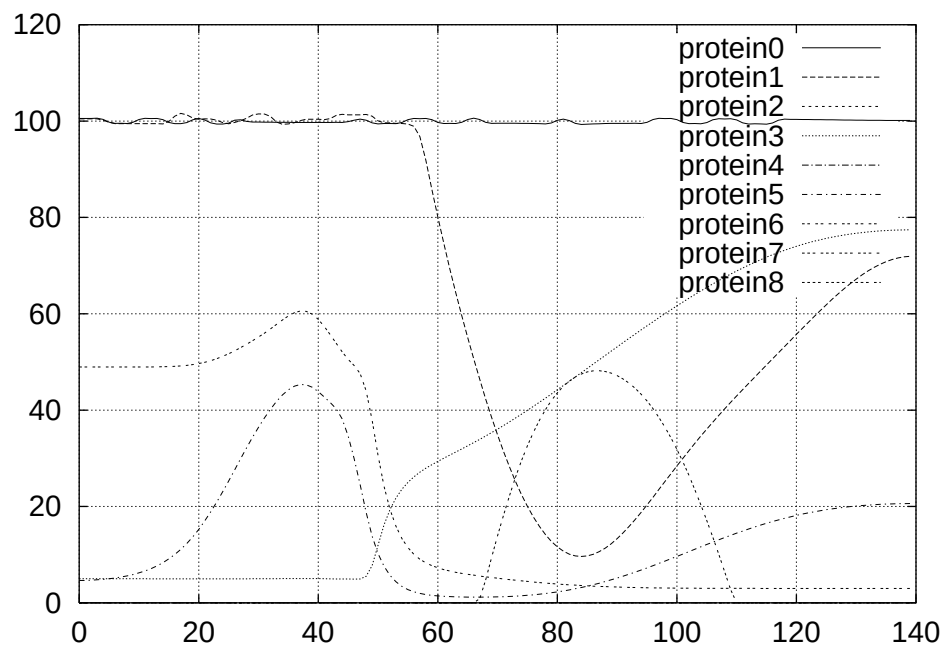


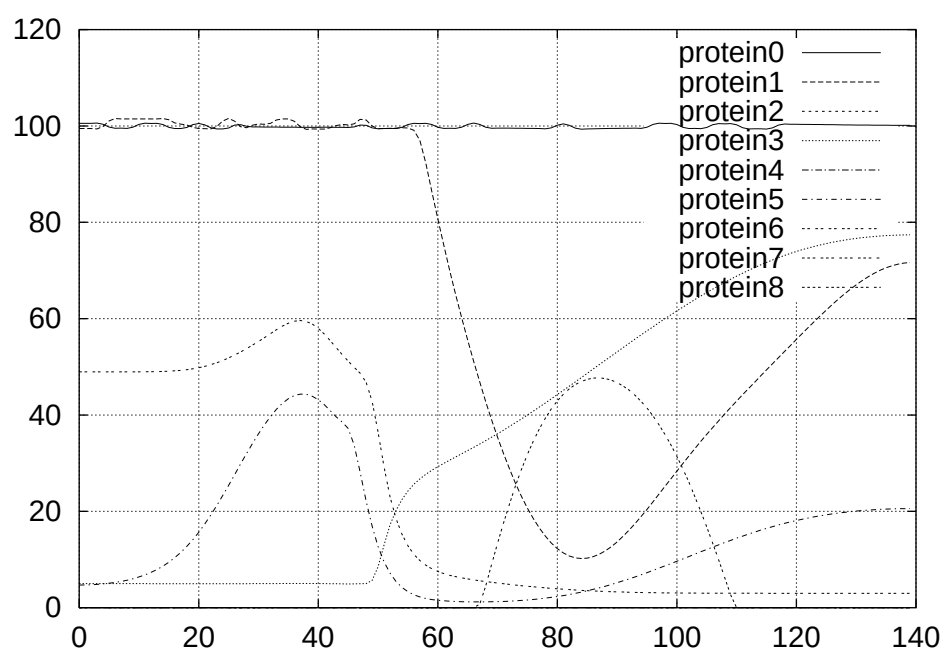
実験3の「拡散-合成-削除」ステップ320の各遺伝子調節タンパク質の分布



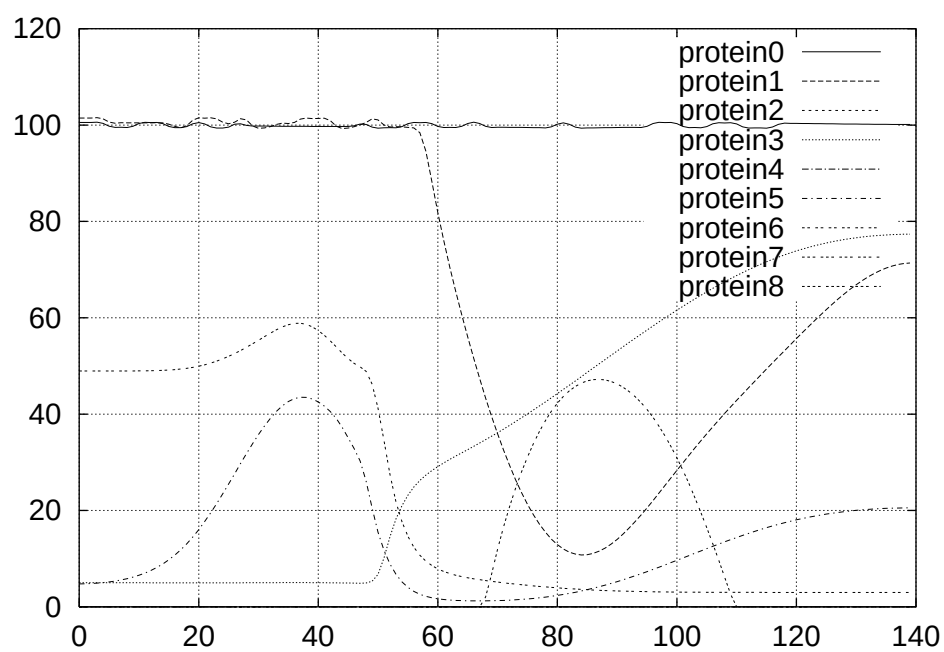


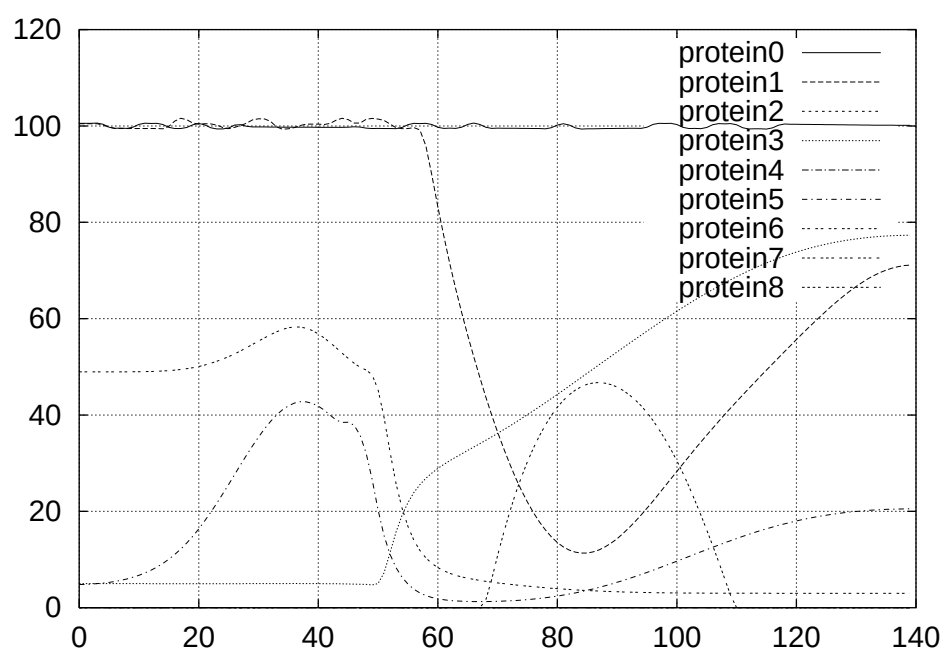
実験3の「拡散-合成-削除」ステップ340の各遺伝子調節タンパク質の分布



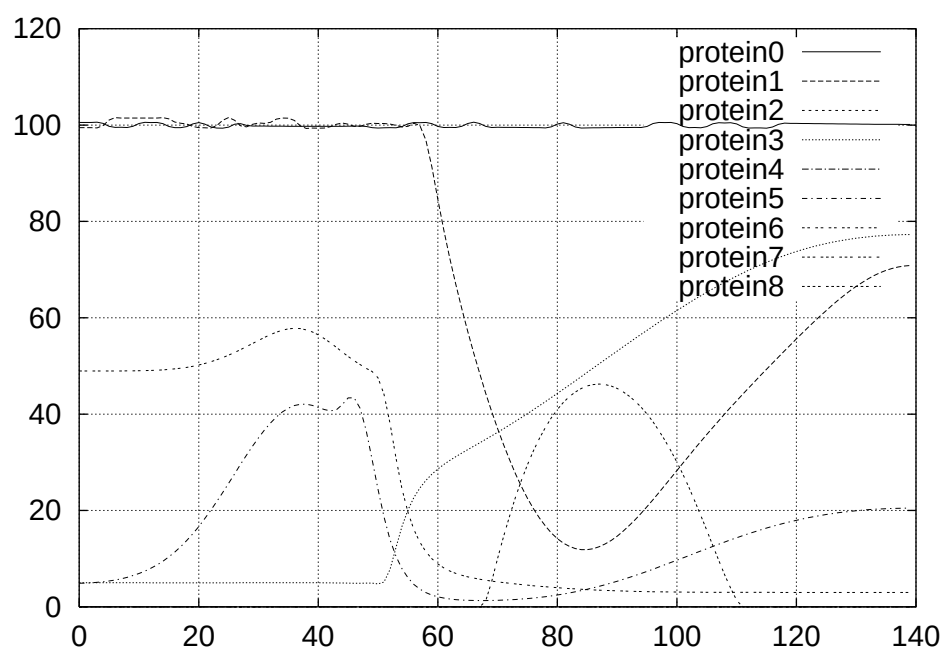


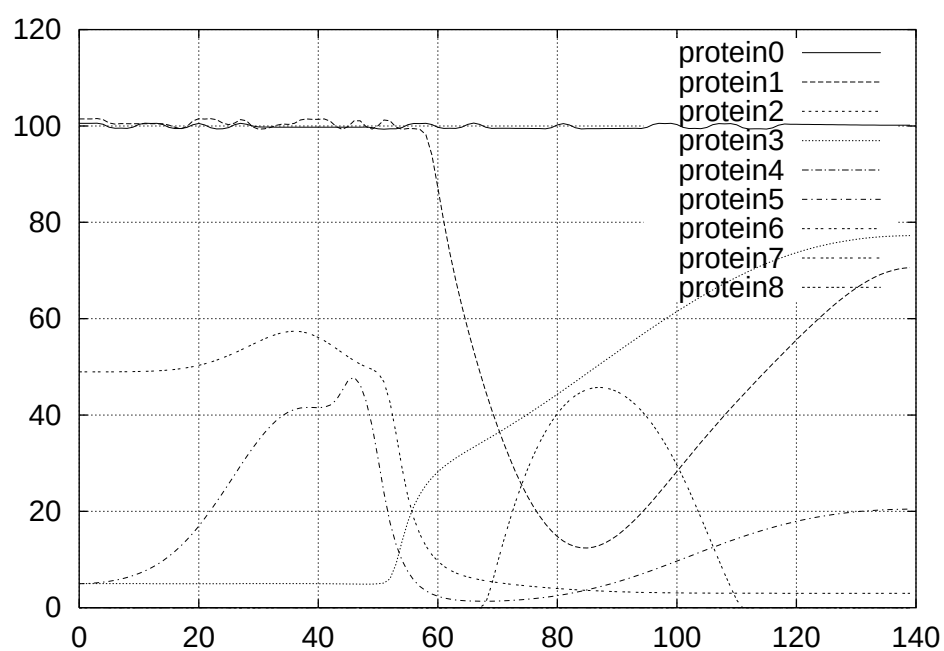
実験3の「拡散-合成-削除」ステップ360の各遺伝子調節タンパク質の分布



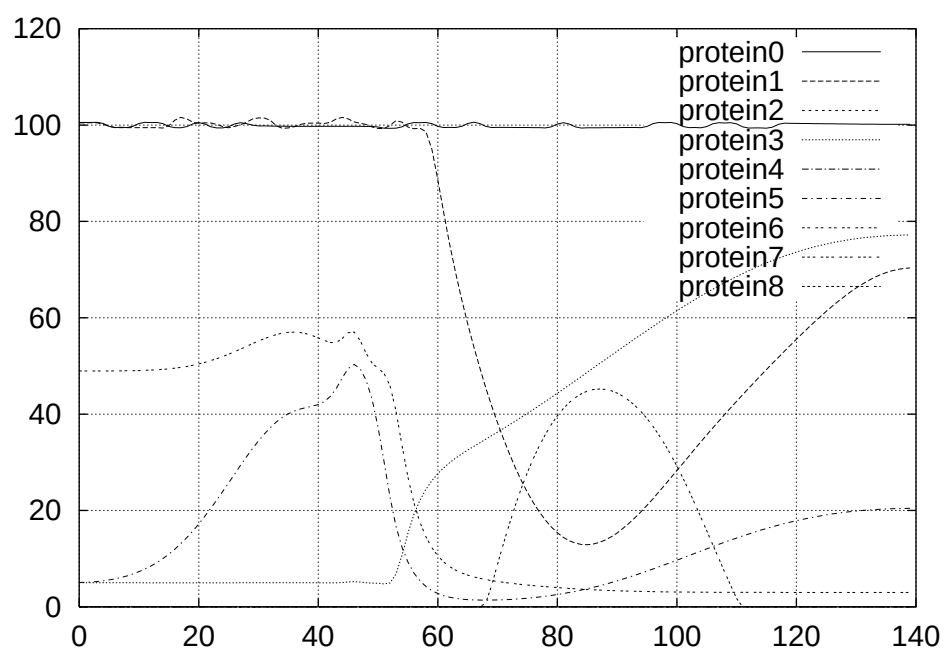


実験3の「拡散-合成-削除」ステップ380の各遺伝子調節タンパク質の分布

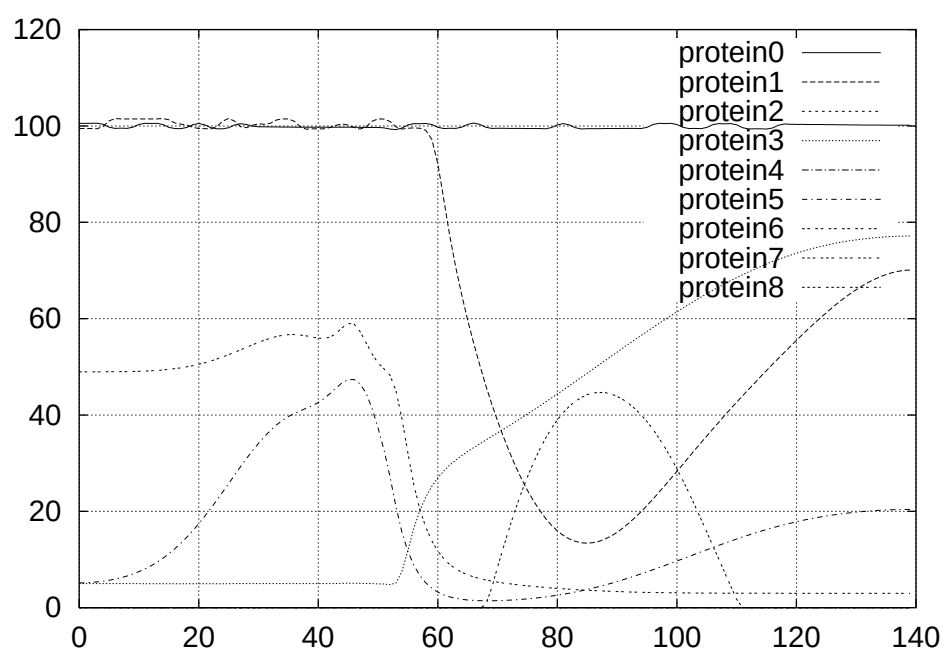




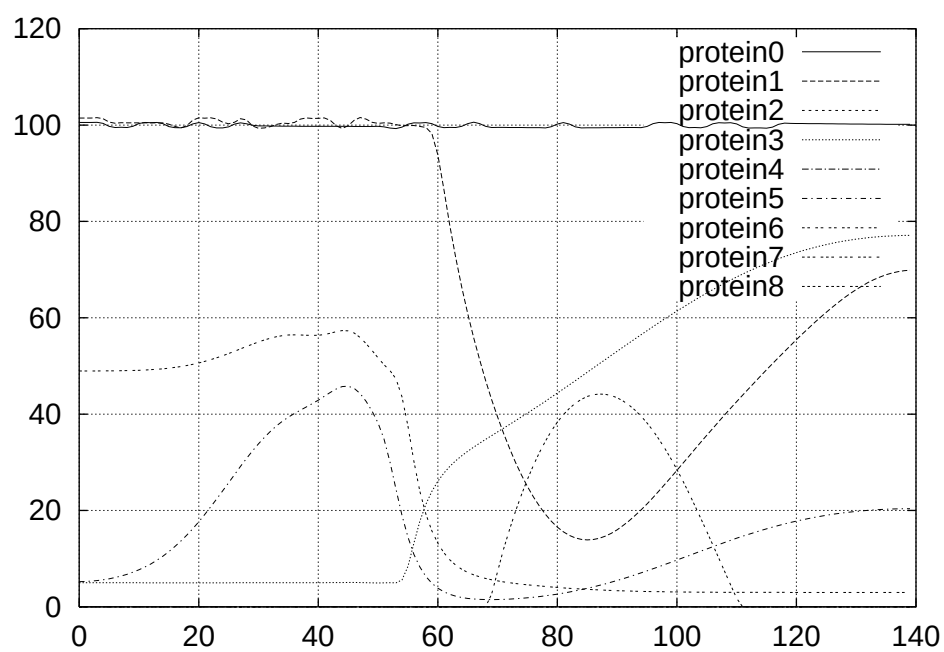
実験3の「拡散-合成-削除」ステップ400の各遺伝子調節タンパク質の分布

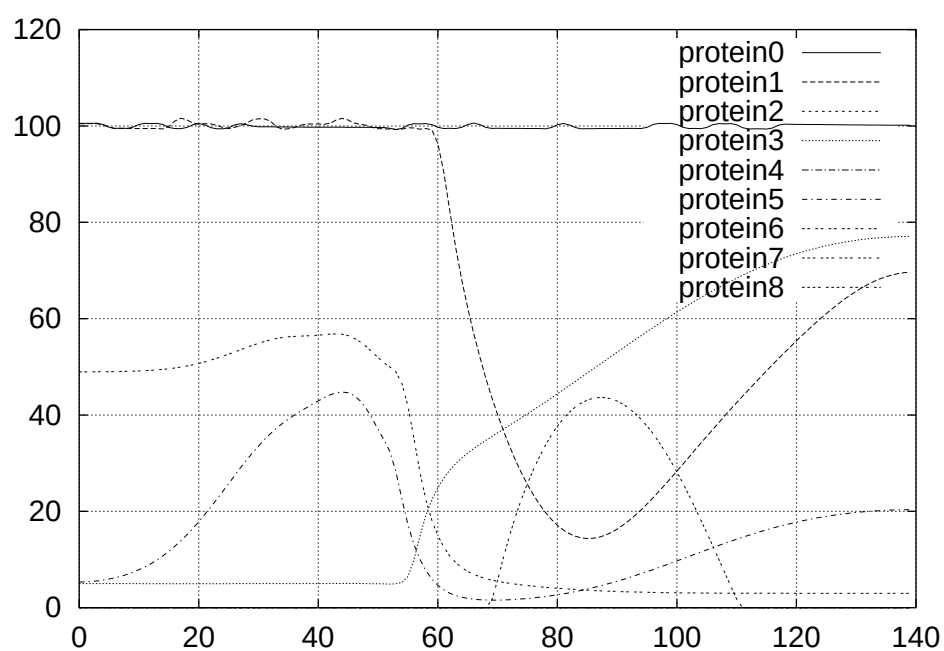




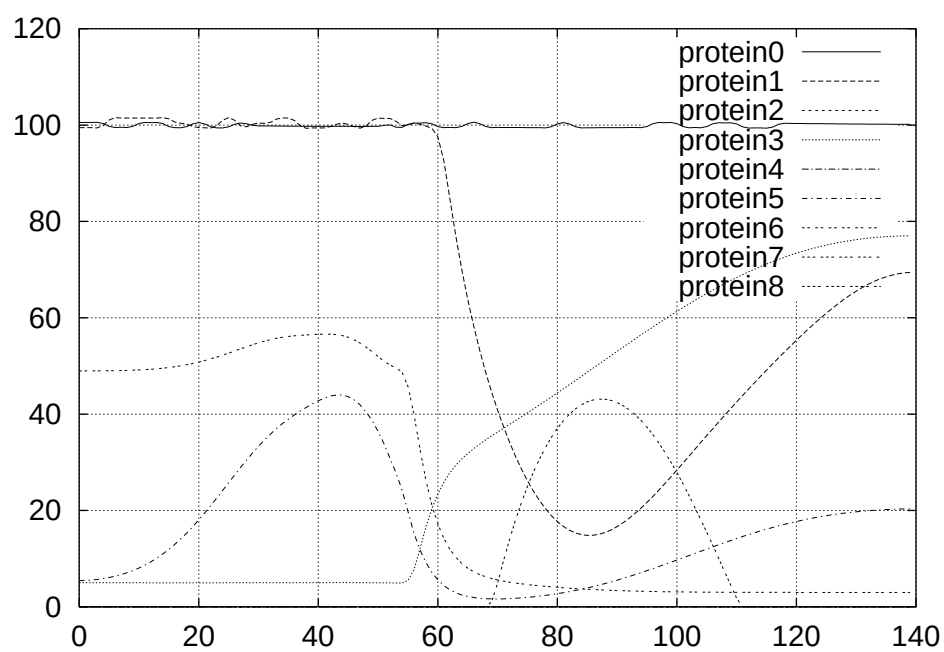


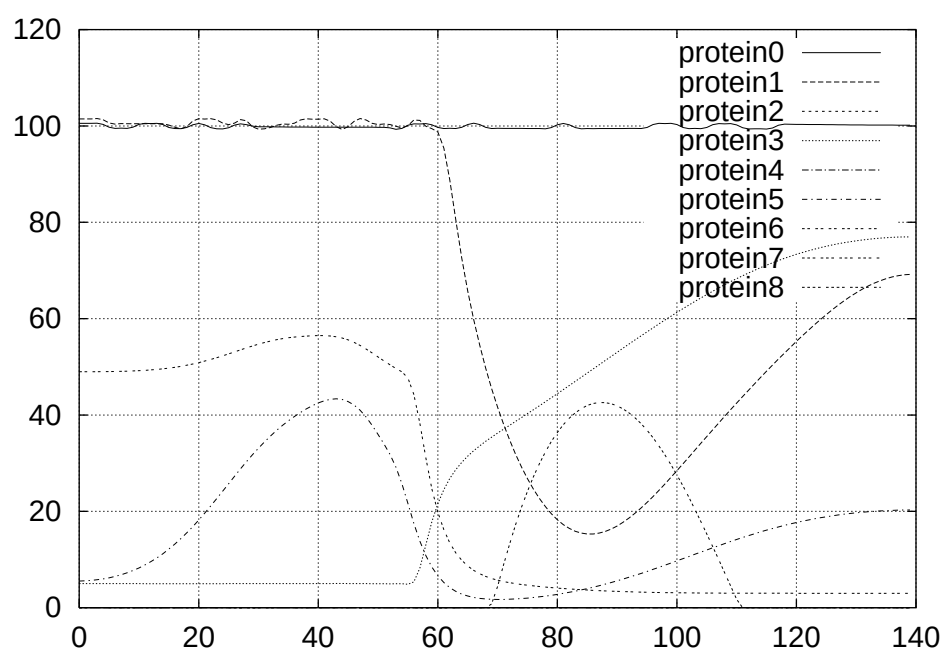
実験3の「拡散-合成-削除」ステップ420の各遺伝子調節タンパク質の分布



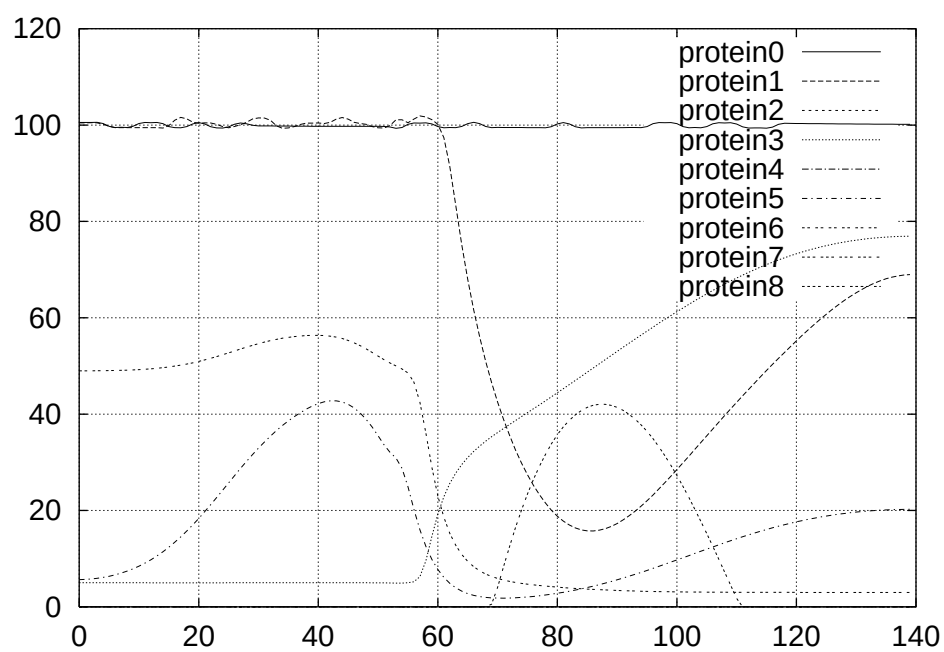


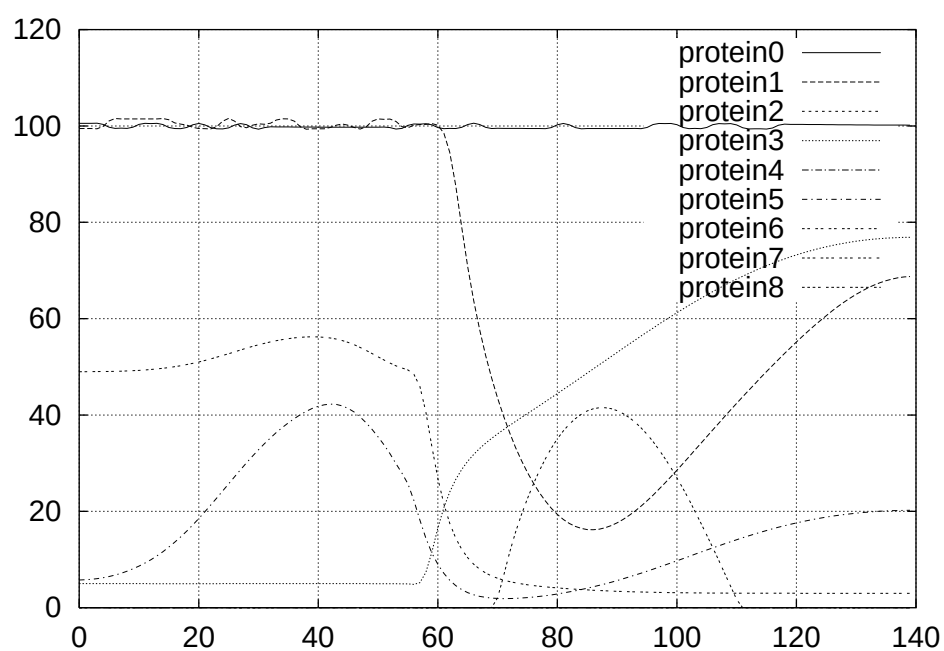
実験3の「拡散-合成-削除」ステップ440の各遺伝子調節タンパク質の分布



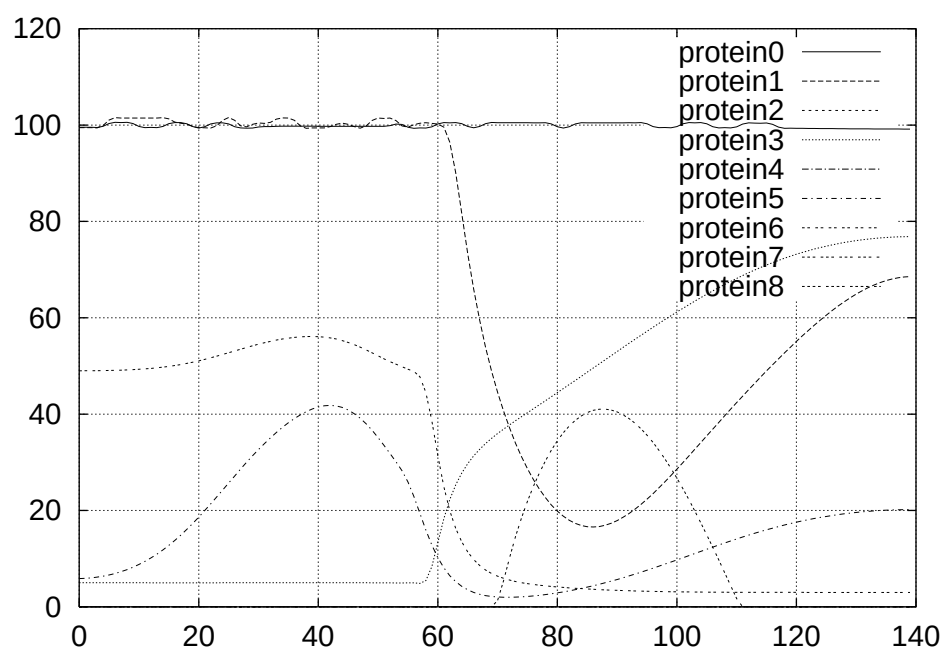


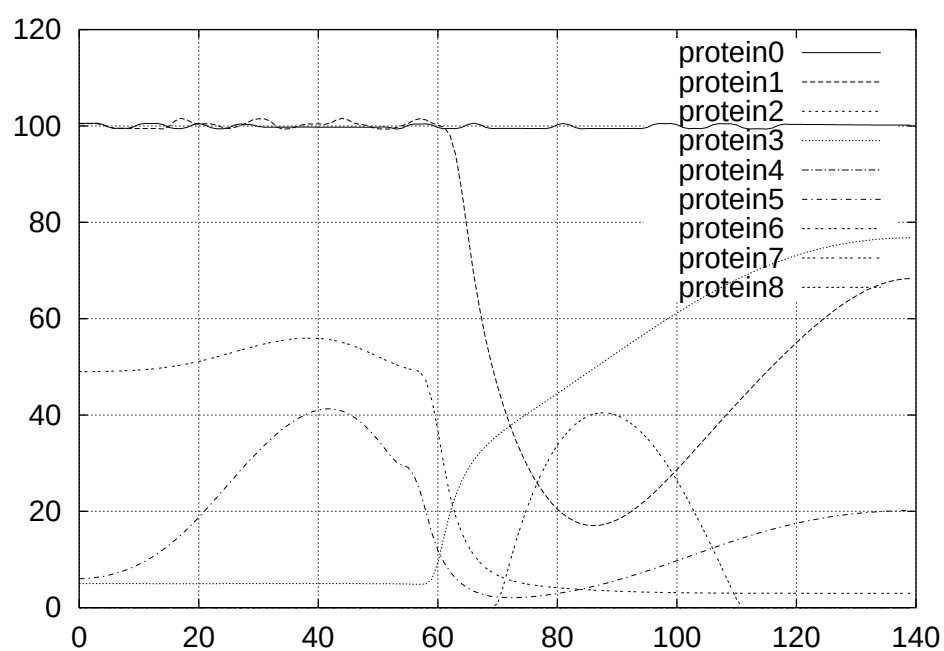
実験3の「拡散-合成-削除」ステップ460の各遺伝子調節タンパク質の分布





実験3の「拡散-合成-削除」ステップ480の各遺伝子調節タンパク質の分布





実験3の「拡散-合成-削除」ステップ500の各遺伝子調節タンパク質の分布

# 付 録 D

## 実験 4

### D.1 実験 4 で得られた最良個体

実験 4 で得られたの最良個体を以下に示す。なお、“protein0” は bicoid、“protein1” は nanos、“protein8” は eve を示す。

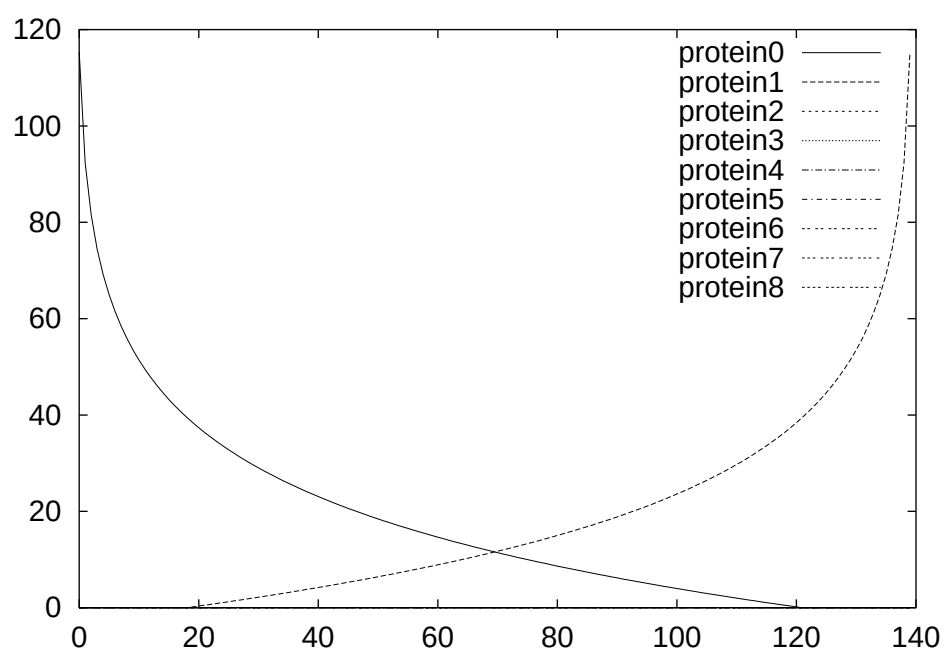
```
(IF (IsConcentrationBetween 1 5.776688 33.021667) protein6)
(IF (IsConcentrationBetween 1 10.724397 6.103053) protein5)
(IF (IsConcentrationBetween 0 10.794619 9.350450) protein8)
(IF (IsConcentrationBetween 0 10.794619 10.025200) protein8)
(IF (IsConcentrationBetween 1 5.307470 37.478039) protein6)
(IF (IsConcentrationBetween 1 17.059525 2.244593) protein4)
(IF (IsConcentrationBetween 1 5.776688 30.096888) protein4)
(IF (IsConcentrationBetween 1 10.724397 6.103053) protein2)
(IF (IsConcentrationBetween 4 28.167505 36.275078) protein8)
(IF (IsConcentrationBetween 2 37.478039 51.445824) protein8)
(IF (IsConcentrationBetween 7 62.635323 0.242824) protein8)
(IF (IsConcentrationBetween 4 2.447898 10.794619) protein7)
(IF (IsConcentrationBetween 1 17.059525 2.244593) protein5)
(IF (IsConcentrationBetween 0 10.794619 9.350450) protein8)
(IF (IsConcentrationBetween 4 11.277724 34.004467) protein8)
(IF (IsConcentrationBetween 4 11.277724 34.004467) protein8)
```

(IF (IsConcentrationBetween 0 9.350450 11.277724) protein7)  
(IF (IsConcentrationBetween 1 10.724397 6.103053) protein7)  
(IF (IsConcentrationBetween 3 32.326157 2.058443) protein6)  
(IF (IsConcentrationBetween 8 38.440750 24.766159) protein2)  
(IF (IsConcentrationBetween 2 37.478039 51.445824) protein8)  
(IF (IsConcentrationBetween 2 37.478039 51.445824) protein8)  
(IF (IsConcentrationBetween 5 20.157558 21.375805) protein8)  
(IF (IsConcentrationBetween 5 20.157558 21.375805) protein8)  
(IF (IsConcentrationBetween 5 20.157558 21.375805) protein8)  
(IF (IsConcentrationBetween 2 37.478039 51.445824) protein8)

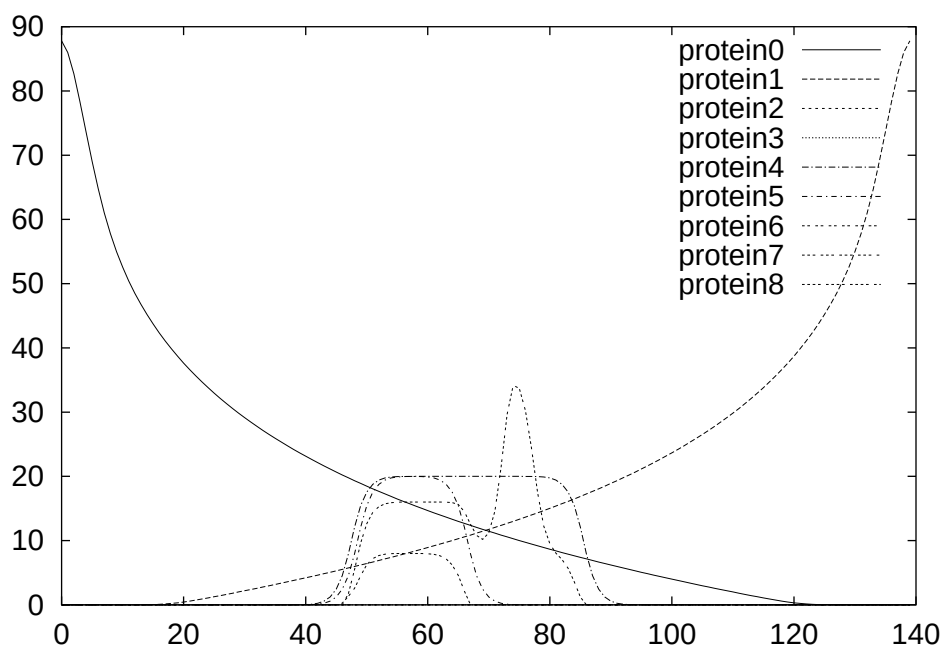
## D.2 最良個体の遺伝子調節タンパク質の分布の推移

実験4で得られた最良個体は、付録D.1に示した通りである。この個体を評価してもっとも正解に近づいたステップは440ステップ目であった。その時の各遺伝子調節タンパク質の分布を図D.2に示す。

以下では、「合成-拡散-削除」ステップごとの遺伝子調節タンパク質の分布の推移を一連の図を挙げて示す。

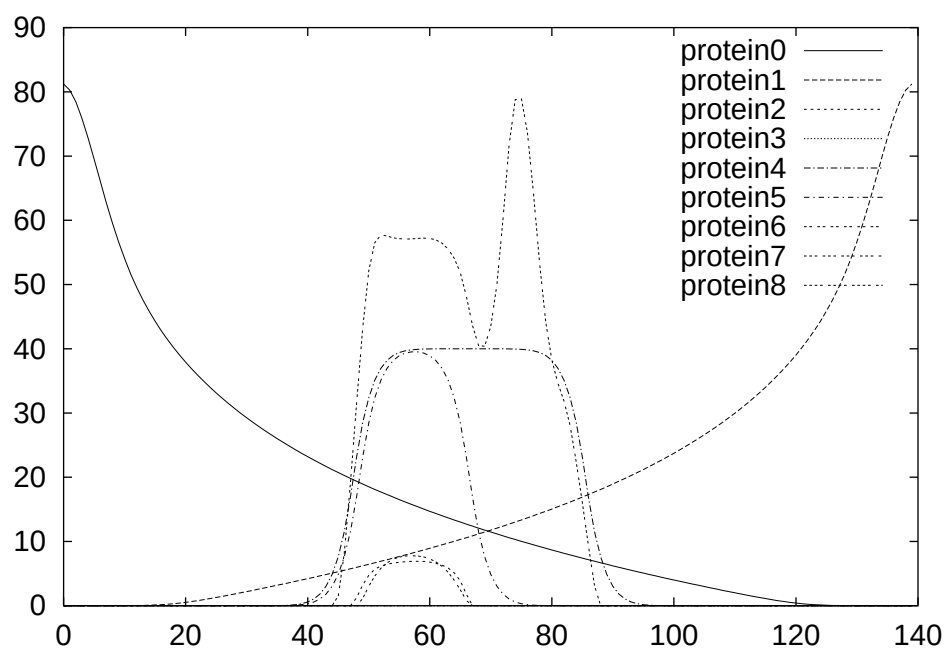


実験4の「拡散-合成-削除」ステップ0の各遺伝子調節タンパク質の分布

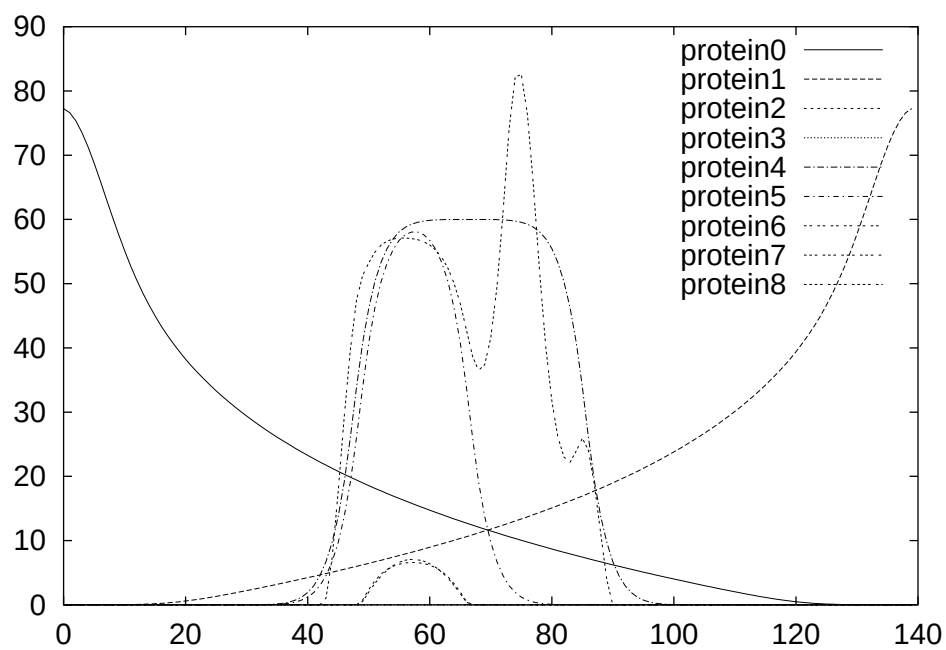


実験4の「拡散-合成-削除」ステップ20の各遺伝子調節タンパク質の分布

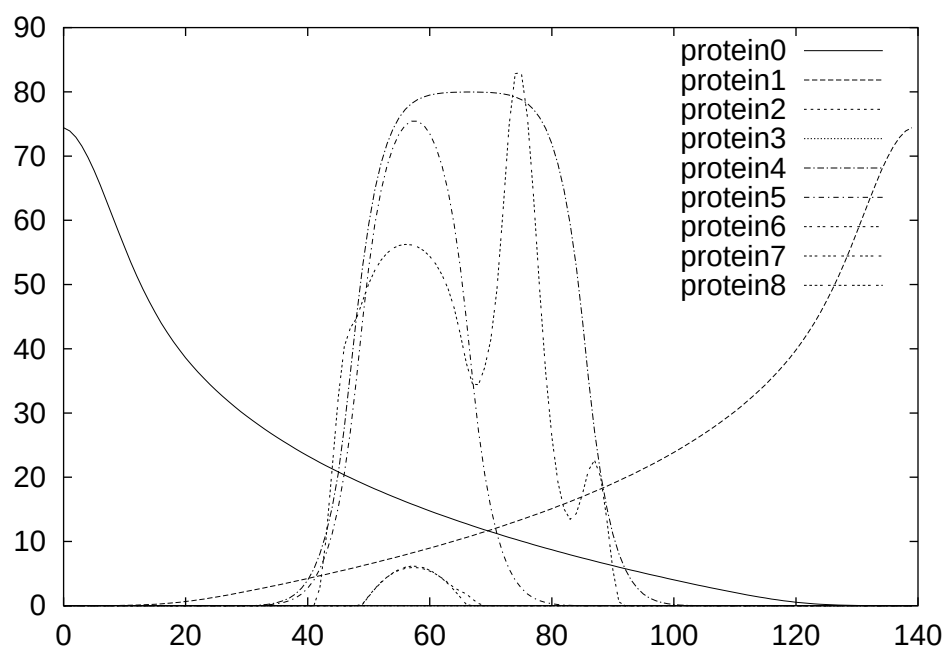




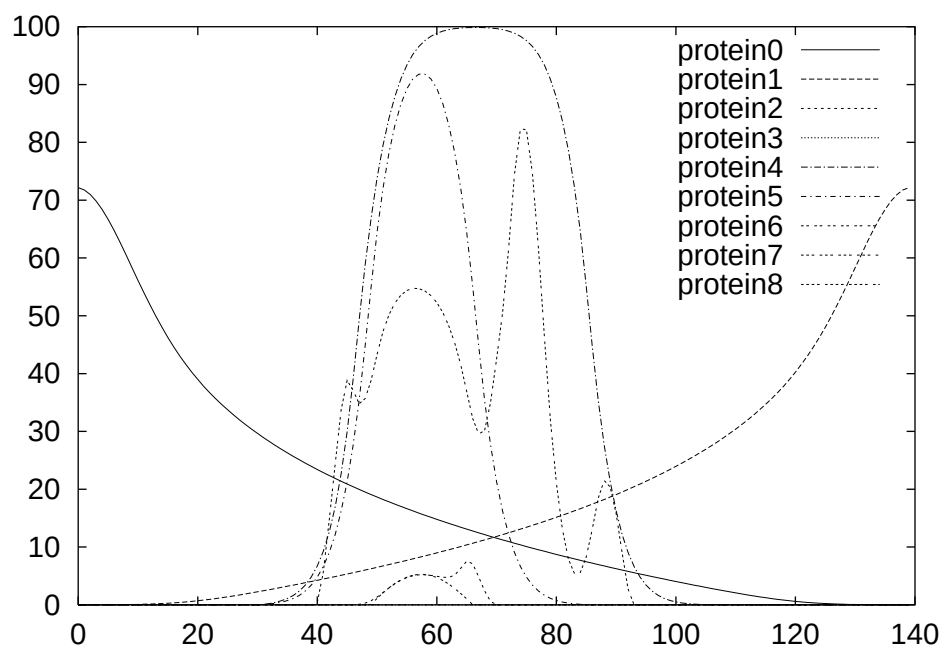
実験 4 の「拡散-合成-削除」ステップ 40 の各遺伝子調節タンパク質の分布



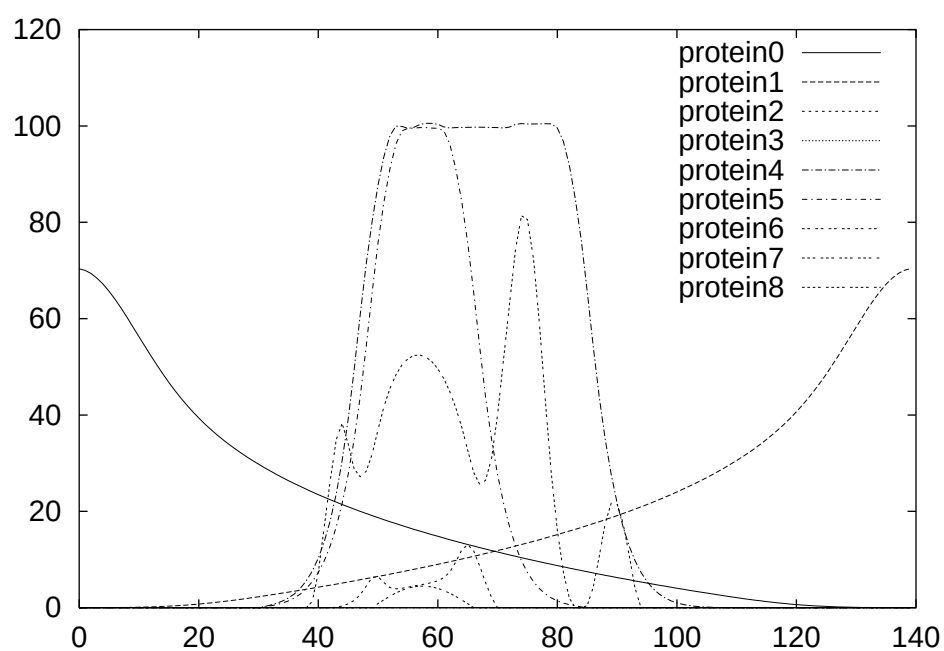
実験 4 の「拡散-合成-削除」ステップ 60 の各遺伝子調節タンパク質の分布



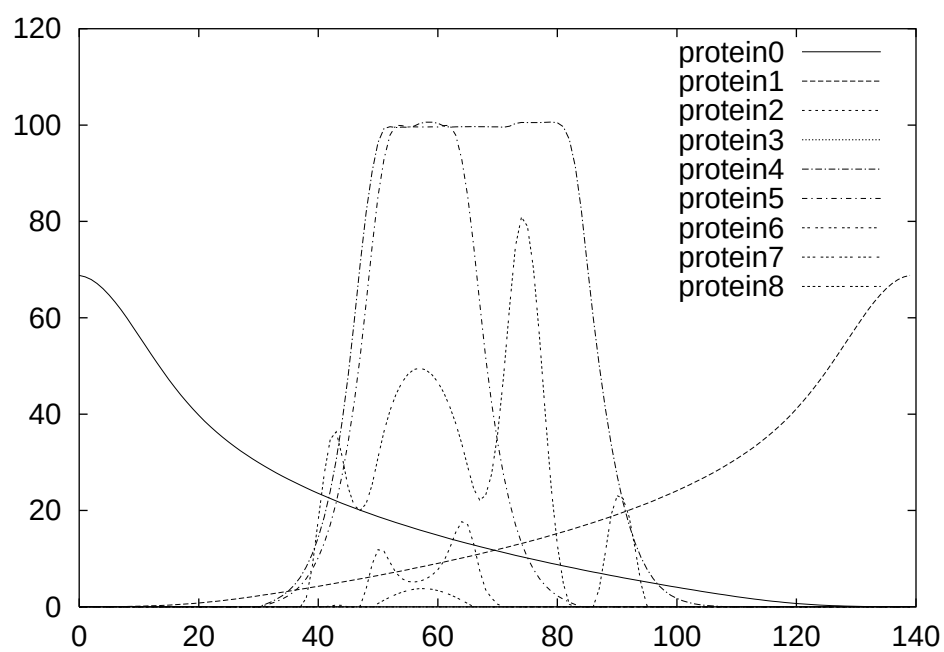
実験 4 の「拡散-合成-削除」ステップ 80 の各遺伝子調節タンパク質の分布



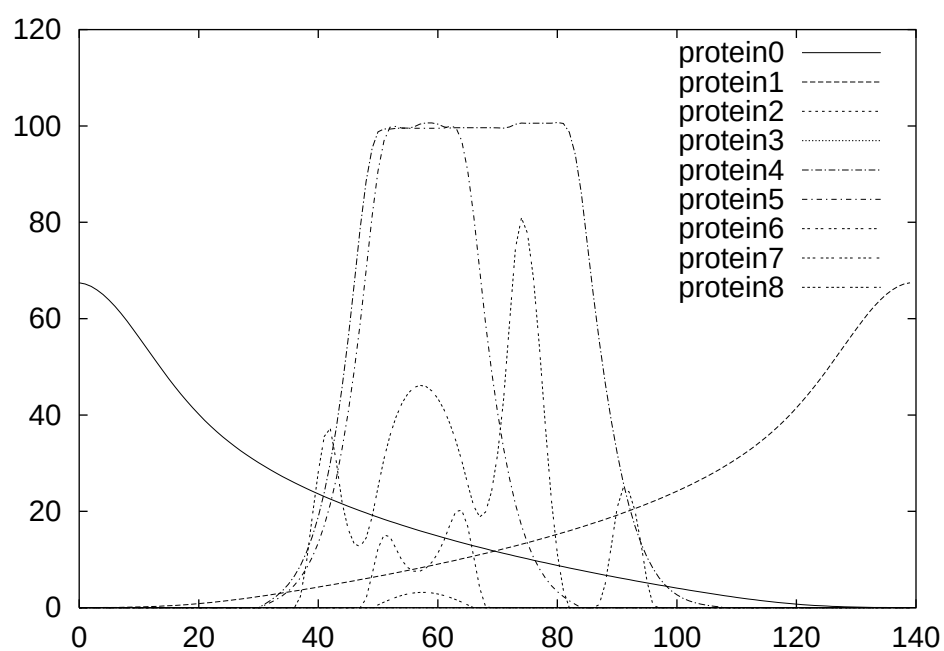
実験 4 の「拡散-合成-削除」ステップ 100 の各遺伝子調節タンパク質の分布



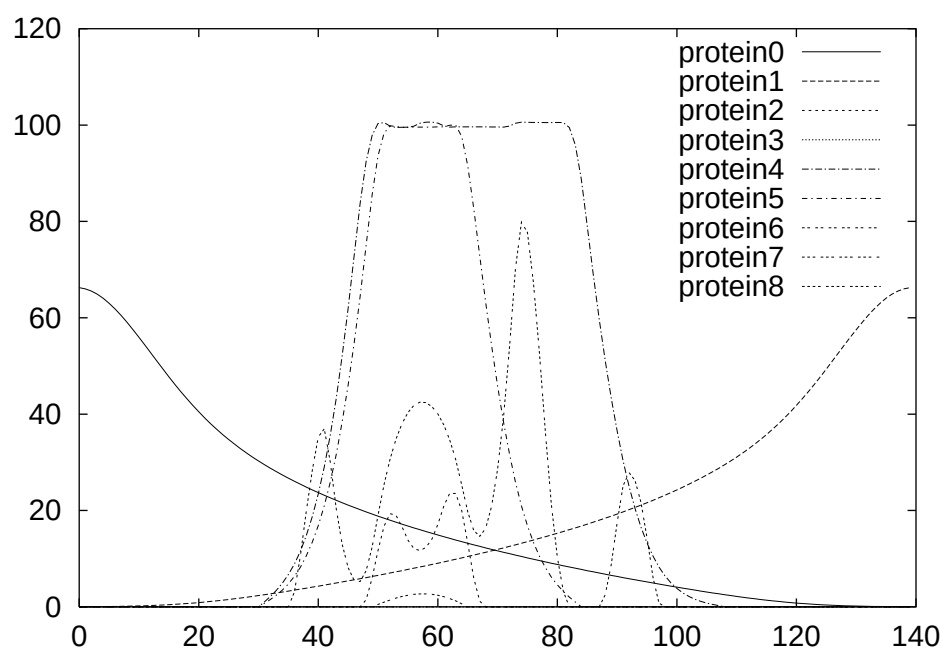
実験 4 の「拡散-合成-削除」ステップ 120 の各遺伝子調節タンパク質の分布



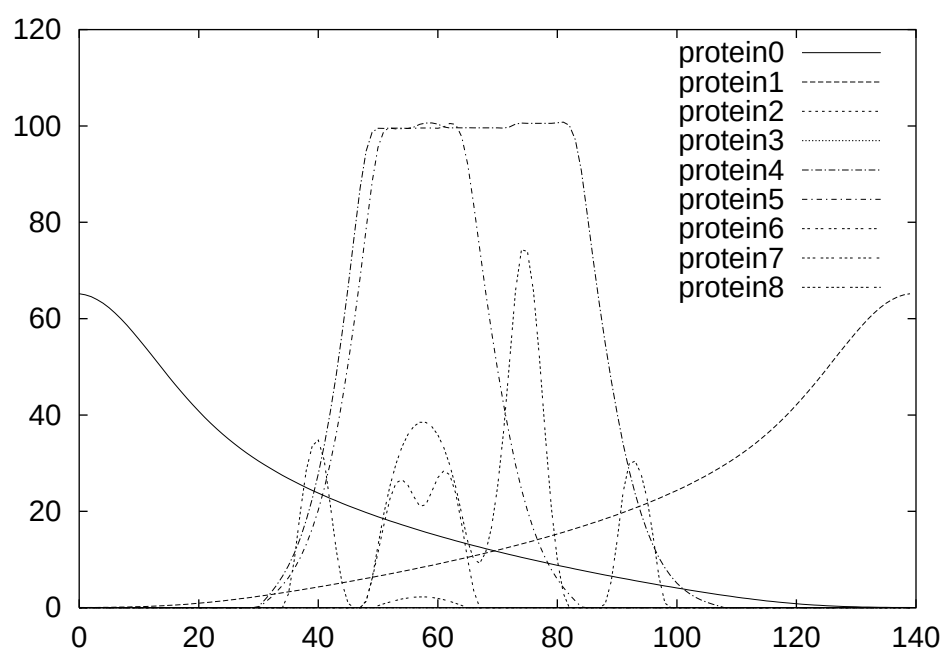
実験 4 の「拡散-合成-削除」ステップ 140 の各遺伝子調節タンパク質の分布



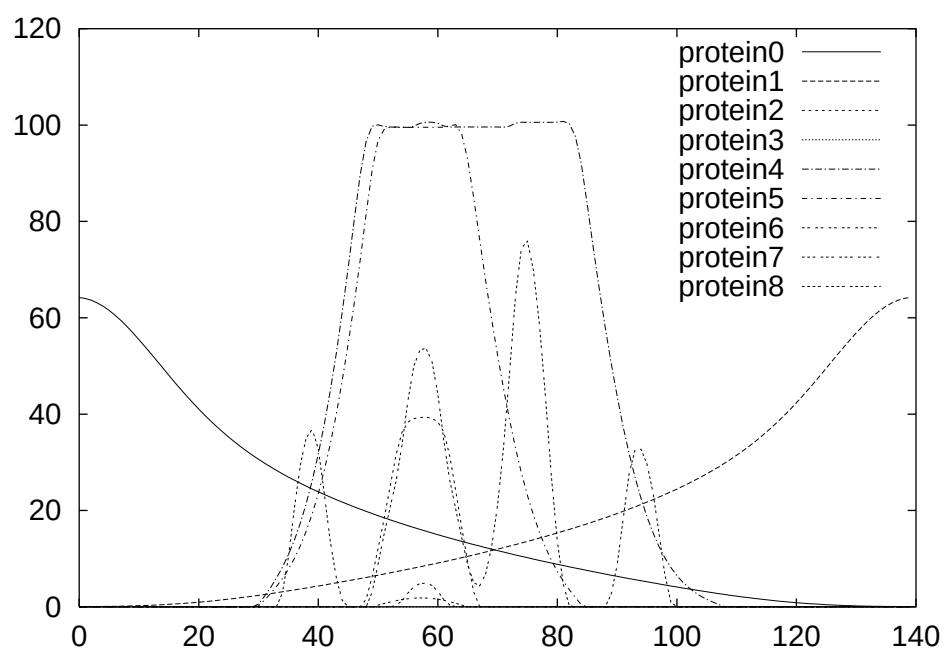
実験 4 の「拡散-合成-削除」ステップ 160 の各遺伝子調節タンパク質の分布



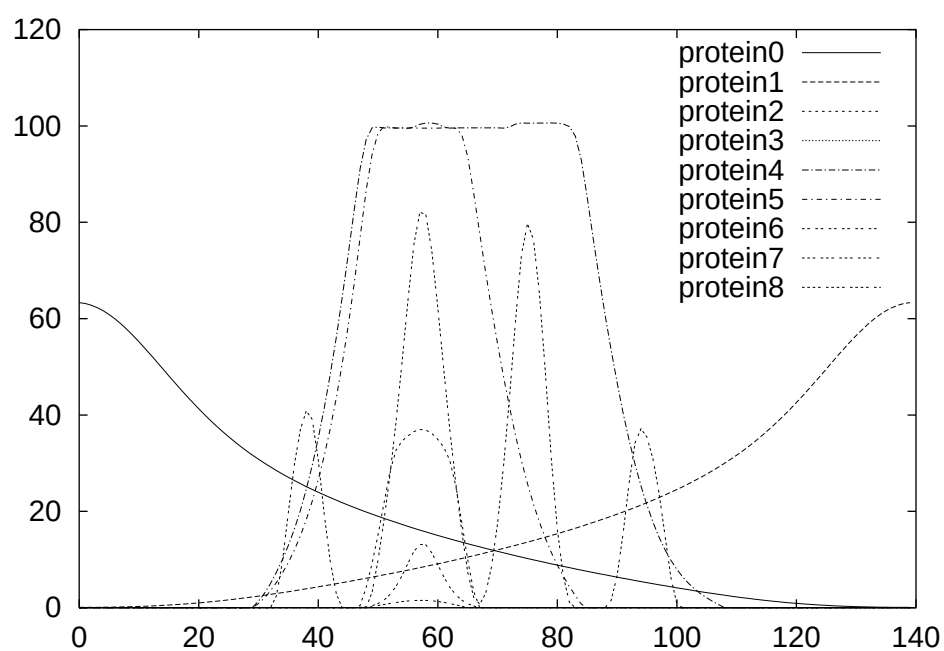
実験 4 の「拡散-合成-削除」ステップ 180 の各遺伝子調節タンパク質の分布



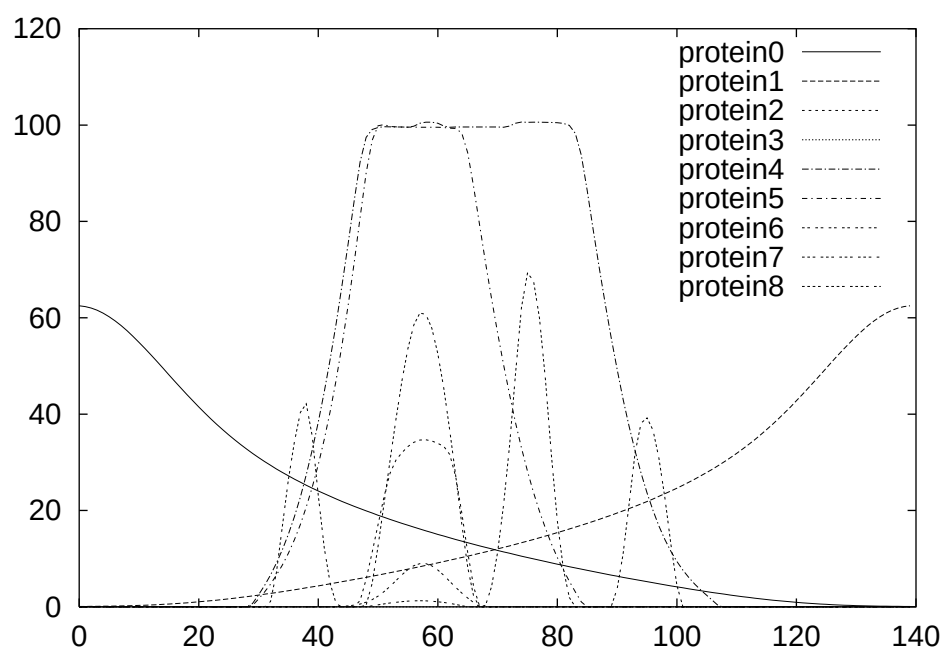
実験 4 の「拡散-合成-削除」ステップ 200 の各遺伝子調節タンパク質の分布



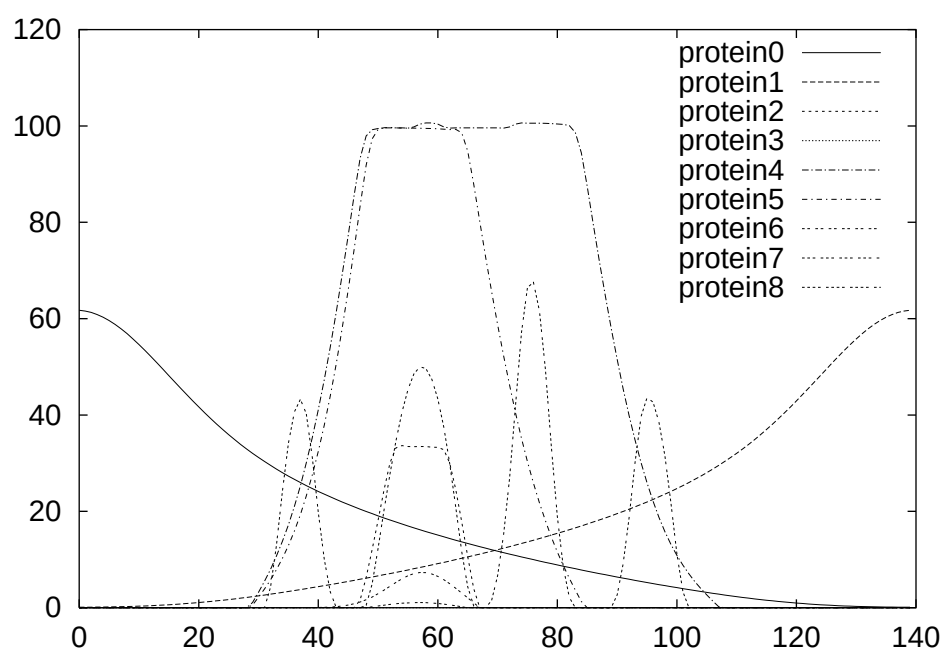
実験 4 の「拡散-合成-削除」ステップ 220 の各遺伝子調節タンパク質の分布



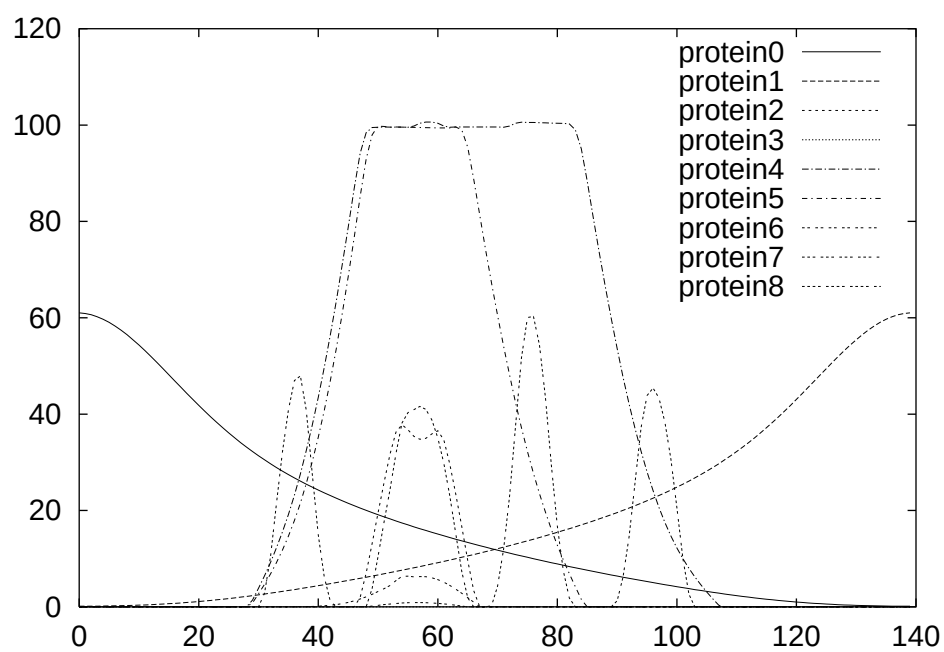
実験 4 の「拡散-合成-削除」ステップ 240 の各遺伝子調節タンパク質の分布



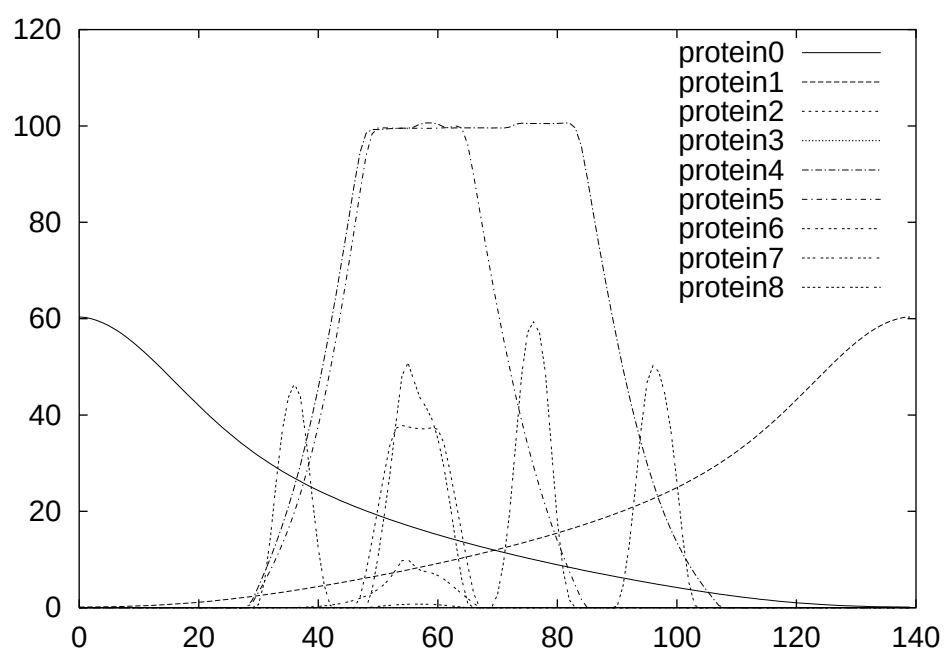
実験 4 の「拡散-合成-削除」ステップ 260 の各遺伝子調節タンパク質の分布



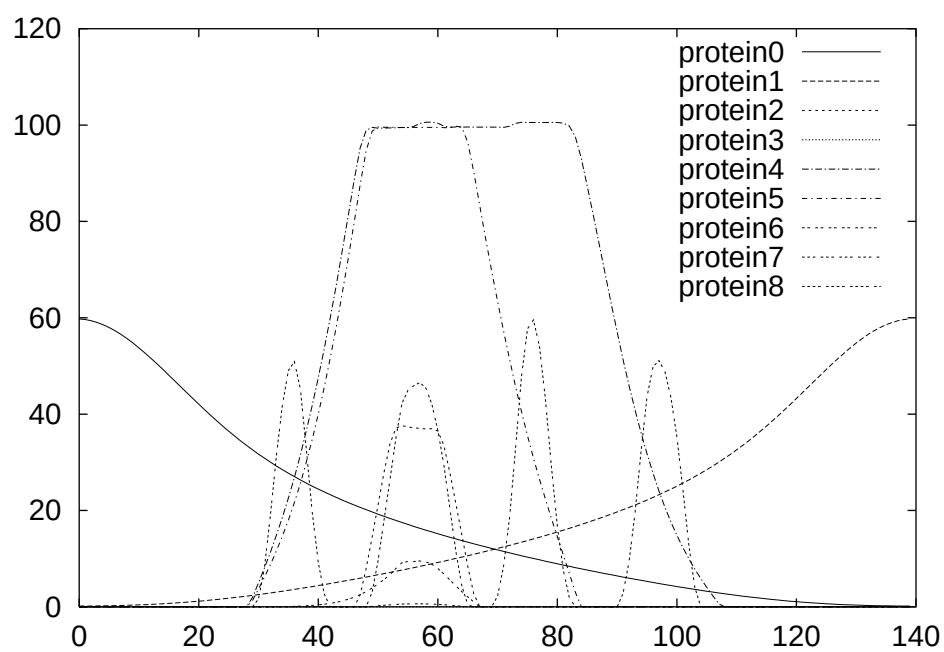
実験 4 の「拡散-合成-削除」ステップ 280 の各遺伝子調節タンパク質の分布



実験 4 の「拡散-合成-削除」ステップ 300 の各遺伝子調節タンパク質の分布

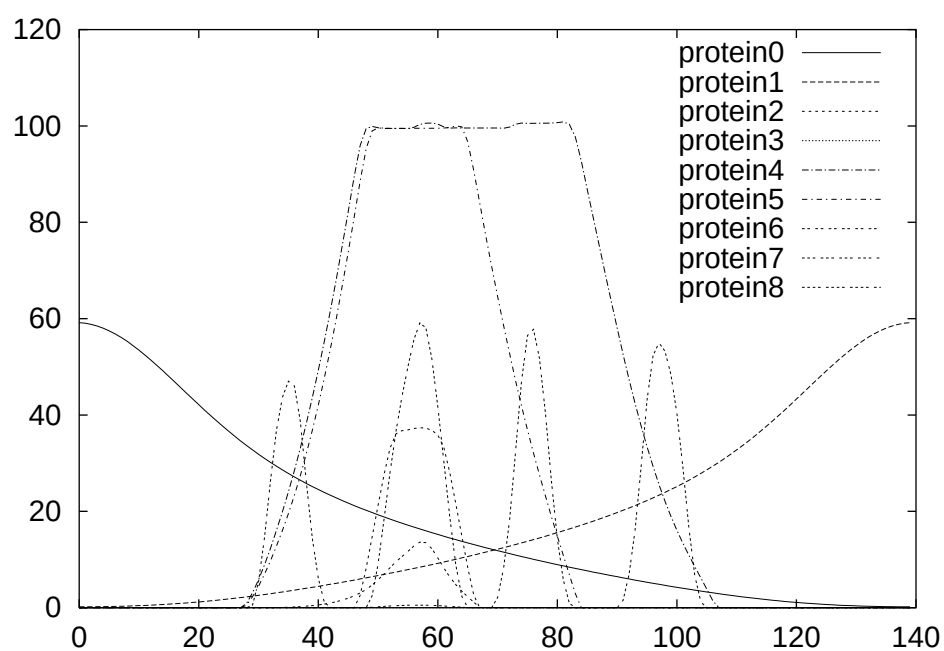


実験 4 の「拡散-合成-削除」ステップ 320 の各遺伝子調節タンパク質の分布

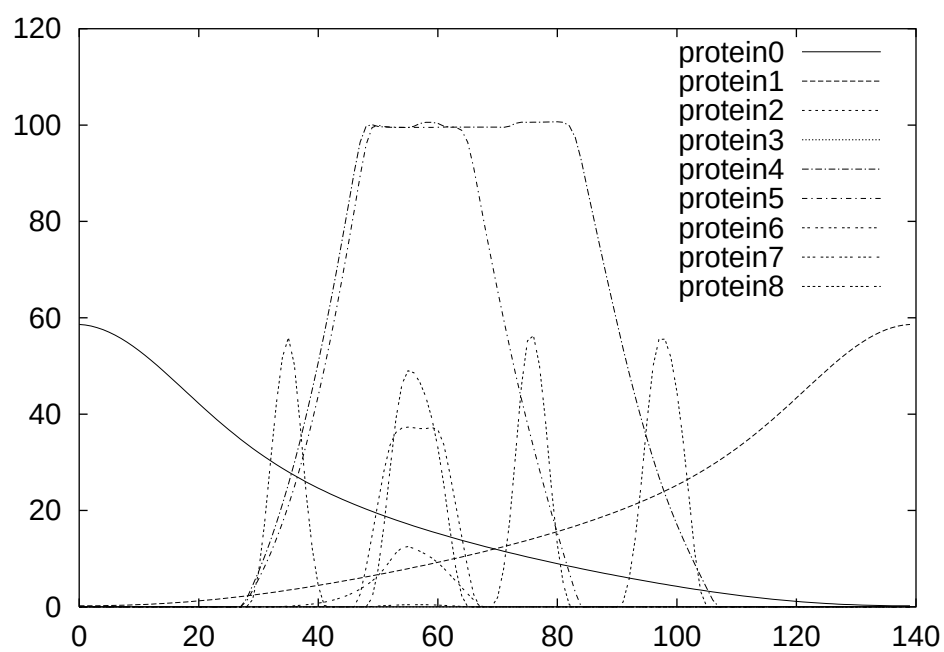


実験 4 の「拡散-合成-削除」ステップ 340 の各遺伝子調節タンパク質の分布

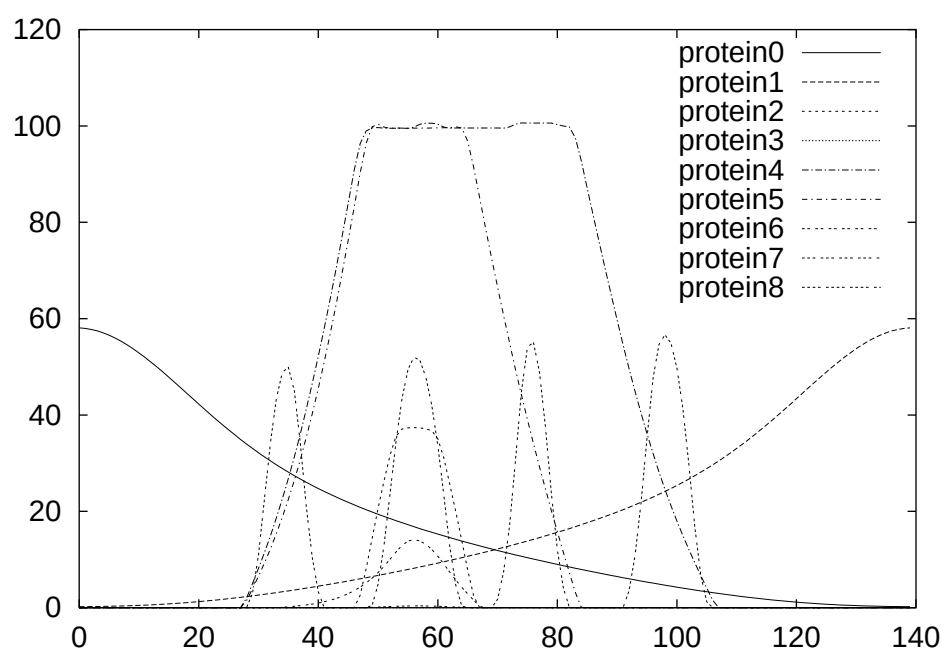




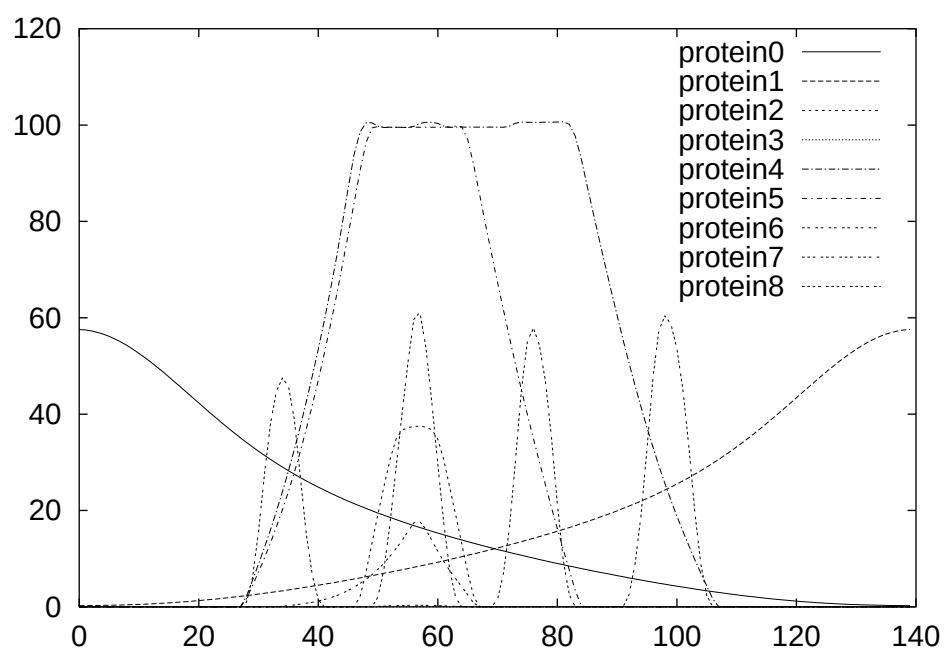
実験 4 の「拡散-合成-削除」ステップ 360 の各遺伝子調節タンパク質の分布



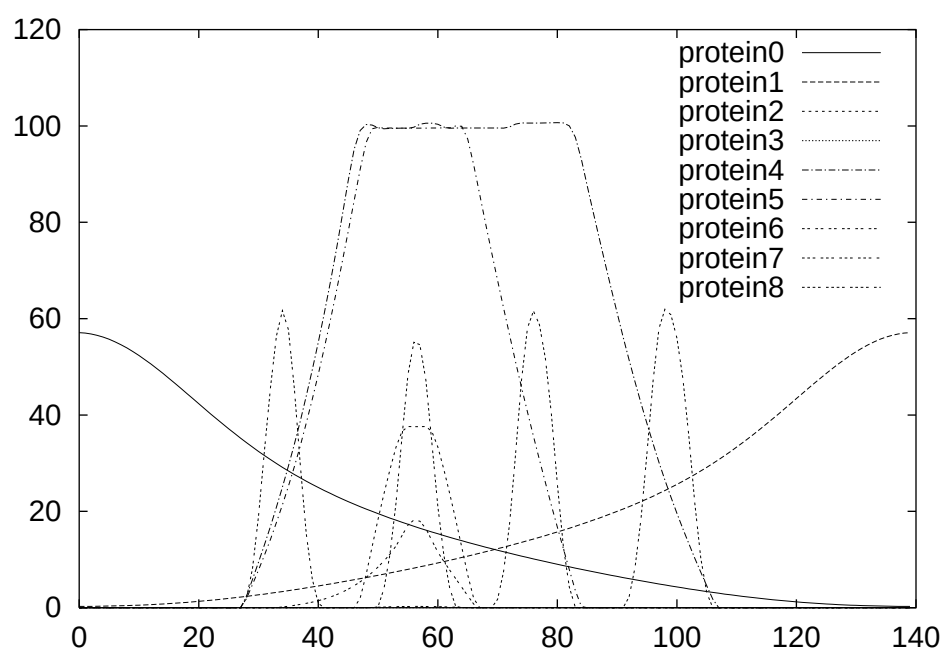
実験 4 の「拡散-合成-削除」ステップ 380 の各遺伝子調節タンパク質の分布



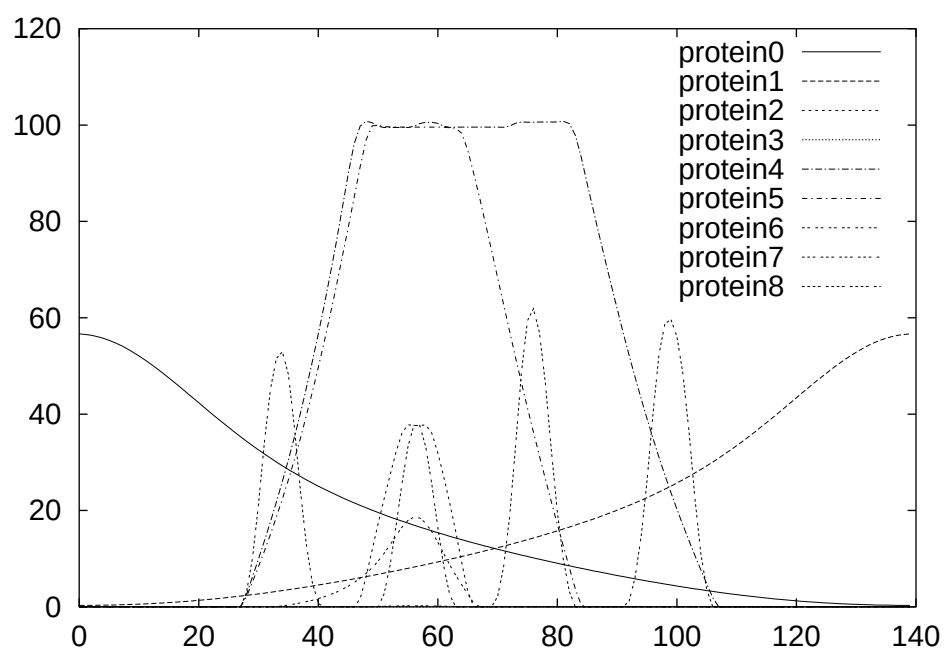
実験 4 の「拡散-合成-削除」ステップ 400 の各遺伝子調節タンパク質の分布



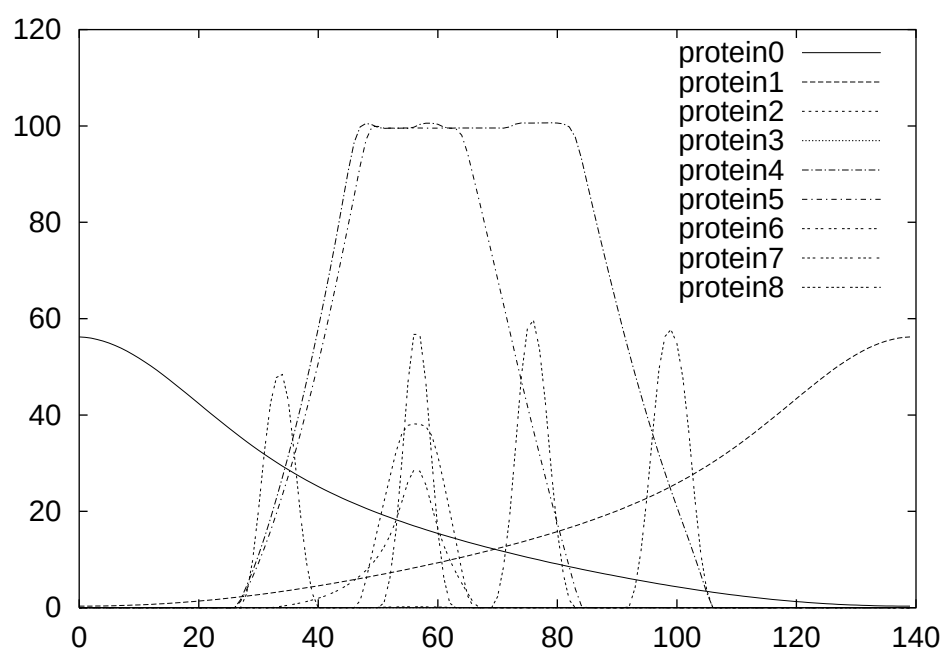
実験 4 の「拡散-合成-削除」ステップ 420 の各遺伝子調節タンパク質の分布



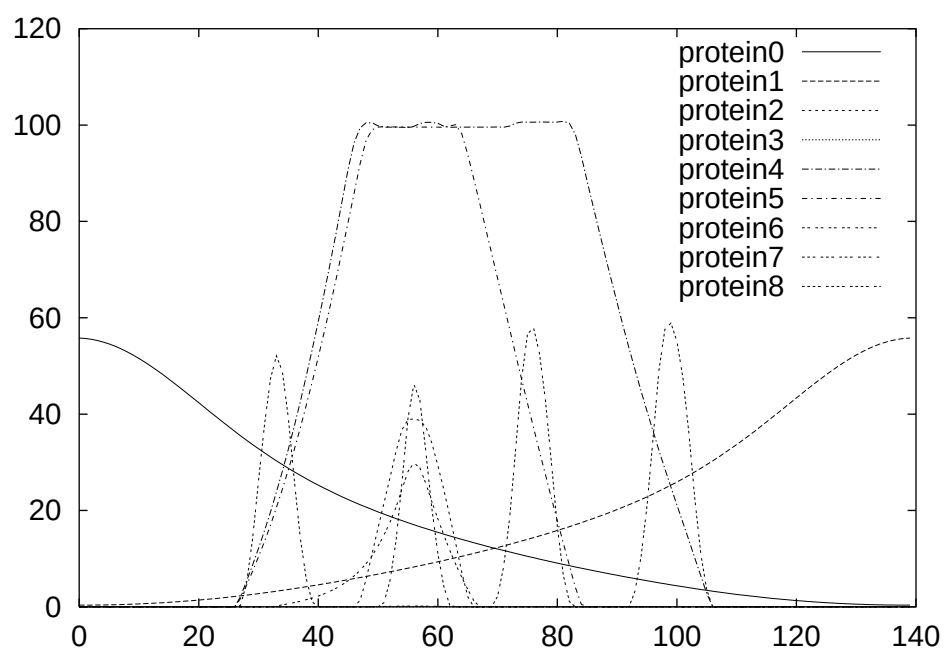
実験 4 の「拡散-合成-削除」ステップ 440 の各遺伝子調節タンパク質の分布



実験 4 の「拡散-合成-削除」ステップ 460 の各遺伝子調節タンパク質の分布



実験 4 の「拡散-合成-削除」ステップ 480 の各遺伝子調節タンパク質の分布



実験 4 の「拡散-合成-削除」ステップ 500 の各遺伝子調節タンパク質の分布